2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

Improving the Generalization of Unseen Crowd Behaviors for Reinforcement Learning based Local Motion Planners

Wen Zheng Terence Ng[1,2], Jianda Chen[1], Sinno Jialin Pan[3], Tianwei Zhang[1]
[1]Nanyang Technological University, [2]Continental Automotive Singapore, [3]The Chinese University of Hong Kong
{ngwe0099, jianda001}@e.ntu.edu.sg, sinnopan@cuhk.edu.hk, tianwei.zhang@ntu.edu.sg

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10610641

***Abstract*— Deploying a safe mobile robot policy in scenarios with human pedestrians is challenging due to their unpredictable movements. Current Reinforcement Learning-based motion planners rely on a single policy to simulate pedestrian movements and could suffer from the over-fitting issue. Alternatively, framing the collision avoidance problem as a multi-agent framework, where agents generate dynamic movements while learning to reach their goals, can lead to conflicts with human pedestrians due to their homogeneity.**

To tackle this problem, we introduce an efficient method that enhances agent diversity within a single policy by maximizing an information-theoretic objective. This diversity enriches each agent's experiences, improving its adaptability to unseen crowd behaviors. In assessing an agent's robustness against unseen crowds, we propose diverse scenarios inspired by pedestrian crowd behaviors. Our behavior-conditioned policies outperform existing works in these challenging scenes, reducing potential collisions without additional time or travel.

I. Introduction

Mobile robots are increasingly used in various applications ranging from industrial automation, service delivery, to agriculture applications [1]. The ability of these robots to maneuver and navigate in complex and dynamic environments is crucial for their successful deployment. One key aspect of mobile robot navigation is *local motion planning*, which aims to find a feasible and safe path for the robot to follow in its immediate vicinity. This task is particularly challenging, as it needs to ensure safe, efficient, and smooth robot movements in the presence of dynamic obstacles (i.e., pedestrians) in the environment. To address this issue, Reinforcement Learning (RL) has been introduced to achieve local motion planning, which exhibits the high ability to handle more complex scenarios and increased levels of uncertainty [2]–[6].

For RL-based methods, the environment is crucial as it shapes the agent's understanding of the environment to train an optimal policy. Particularly, the scenes in the environment should fully reflect the inherent diversity and unpredictability of pedestrians' movements. For example, on the footpaths, human pedestrians may walk at different speeds or behave differently depending on their social norms. If their behaviors are not modeled comprehensively, it is challenging for the robot agent to learn a robust policy which works well against diverse and unseen crowd behaviors.

Various approaches have been proposed to generate pedestrian movements for training RL-based local motion planning policies, which can be classified into two categories. Unfortunately, both suffer limitations. The first category involves single-agent approaches. One simple approach is to assign pedestrians' waypoints from a dataset [7], which doesn't enable interactions between robots and pedestrians, limiting their influence on each other. To address this, some works manually design pedestrians' behaviors based on crowd density [8], [9], or use fixed non-learning-based algorithms to control pedestrians [4], [10]–[12]. However, these approaches may lead to agents overfitting due to limited diversity in generated pedestrian movements.

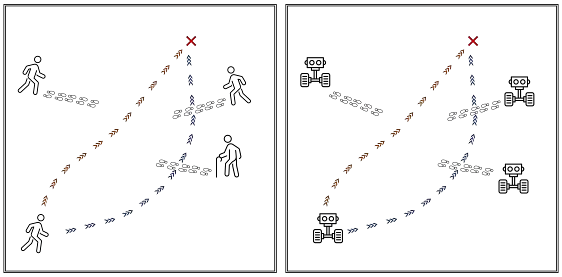

Fig. 1: **A human may take diverse strategies to reach the same predefined goal (left). We propose a behavior-conditioned policy to integrate such diversity into the robot agent (right). This diversity enriches the agent with a more varied range of experiences when learning in a multi-agent framework, and improves its ability to generalize in unseen crowd behaviors.**

The second approach frames local motion planning as a decentralized multi-agent collision avoidance task [3], [5], [13]. Each agent is linked to a single policy to reach its goal, learning to avoid collisions and adapting dynamically during training. This strategy offers two advantages: (1) it doesn't require explicit specification of each pedestrian's behavior to avoid bias; (2) it's highly sample-efficient due to bootstrapping, as every agent's trajectory can update the same model. However, these solutions face practical challenges when applied to scenarios with diverse or unforeseen dynamics. The learned policy assumes homogeneous behaviors among multiple robots upon deployment, posing a challenge for navigating in scenarios with varied crowd behaviors where such assumptions don't hold.

In this paper, we propose a novel sample-efficient multi-agent framework to enhance behavior diversity among agents. Diverse movements induced by different agents enrich experiences and enhance robustness to unpredictable behaviors in unseen or challenging scenarios. Our framework introduces the concept of behaviors conditioned on a policy. These behaviors, randomly sampled as token embeddings by each agent at the start of each training episode, incentivize agent diversity. We assign intrinsic rewards for agents to take varied actions for every state conditioned on the sampled behavior. These rewards are based on a discriminator capable of identifying behavior from a state-action pair. With this approach, we can train robust RL policies for local motion planning in highly complex environments.

We perform comprehensive simulation experiments to evaluate the robustness of our framework on a diverse set of unseen pedestrians' behaviors. Simulation results show that the proposed behavior-conditioned policy is more robust while having the same number of updates.

II. Preliminaries

A. RL-based Local Motion Planning

Local motion planning is a sequential decision-making task that can be formulated as a Markov Decision Process (MDP), defined by a tuple $M = \langle S, A, P, R, \gamma \rangle$. Here, S is the state space, A is the action space, P is the state-transition model, R is the reward function, and γ is a discount factor.

A general form of the states is $s = [s_{env}, s_{robot}, s_{goal}]$, where s_{env}, s_{robot} and s_{goal} contain information about the observed environment, robot and goal respectively. Similar to [5], we choose a realistic representation that uses distance readings from a 2D laser range finder to sense the environment s_{env}. We consider the sensor noise and obstacle occlusions, and make no assumptions about the shape, size, and number of obstacles, which are more closely aligned with the real world. s_{robot} reveals the state of the robot, usually the velocities and optionally the position. s_{goal} is represented by either the relative or absolute goal position.

The action space A is the set of permissible velocities in either the continuous or discrete space. The reward R can be normally represented as follows:

$$r_t = \begin{cases} r_{goal} & \text{if } \|\mathbf{p}_t - \mathbf{g}\| < d_{col}, \\ r_{col} & \text{else if collision }, \\ r_{step} \cdot (\|\mathbf{p}_{t-1} - \mathbf{g}\| - \|\mathbf{p}_t - \mathbf{g}\|) & \text{otherwise}, \end{cases}$$

where r_{goal} is the reward for reaching the desired goal, r_{col} is the penalty for collision, r_{step} is the dense reward for getting closer to the goal, d_{col} is the distance threshold for reaching the goal, $\mathbf{p}$ and $\mathbf{g}$ are positions of the robot and goal.

In this MDP, we aim to use RL to find a policy π_θ parametrized by θ, which maps states to actions and maximizes the expected sum of discounted rewards, $J(\theta) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \cdot r_t\right]$, where T is the length of an episode.

B. Pedestrian Modelling

In local motion planning, the movement of dynamic obstacles, often represented by human pedestrians, is a crucial environmental factor. Each pedestrian's behavior significantly influences the environment and how the robot agent learns within the RL framework. Existing approaches to modeling pedestrians can be classified into two categories.

Single-agent approaches. In these approaches, only a single robot agent learns to navigate within the crowd, while pedestrians are typically modeled using non-learning-based algorithms. Some examples of the non-learning-based algorithms include Velocity Obstacles [14], Social Forces [4], [10]–[12] and physics inspired movements [15]. One common drawback to these approaches is that the RL-agent might overfit to the chosen pedestrian behaviors during training.

Multi-agent approaches. The main idea of the multi-agent framework [3], [5], [13], [16]–[20] is to control multiple agents with a single policy. In this setup, agents must learn to avoid each other to reach their goals, leading to the emergence of dynamic movements during the course of training. One consequence under this framework is that the agents converge to homogeneous behaviors as all agents boot-strap to a single policy.

C. Behavior Diversity in RL

To address the above homogeneity concern, various approaches were proposed to increase agent behavior diversity. For single-agent scenarios, one popular solution [21] is to maximize the entropy of the policy in addition to the reward, to learn different behaviors to achieve the goal . Eysenbac et al. [22] introduced DIYAN to increase the diversity of agent behaviors by maximizing the mutual information between skills and states, resulting in better state exploration.

In the multi-agent scenario, some work increases the behavior diversity of multiple agents in the Centralized Training with Decentralized Execution (CTDE) framework [23]–[25]. When agents are assigned different tasks, each agent gets a distinct policy respectively, which shares a common critic network. With multiple policies, several ideas have been proposed to generate diverse behaviors among agents [26]–[28]. However, this comes at the expense of sample efficiency since each agent only updates its own policy instead of a unified policy.

III. Approach

We present our approach to learning robust agents through behavior diversity. Instead of using multiple policies to create diversity as in CTDE, we opt for a more sample-efficient method by using only a single policy. We first formulate the agent behaviors, and how they can be used to generate diversity among agents (Section III-A). Then we explain how the behaviors and diversity can be integrated together within a single policy (Section III-B). Finally, we describe how to train a behavior-conditioned policy in an end-to-end manner with all the integrated components (Section III-C).

A. Agent Behavior

In Figure 1, people's approaches to walking towards a goal can vary: some prioritize speed with a longer path, while others choose a shorter route at a slower pace. When avoiding moving obstacles, some turn left, while others turn right. Although individuals may have unique behaviors, there can be similar patterns. We formalize this with discrete behavior tokens $z \in [0,1,\dots,M-1]$, where M is the total number of distinct behaviors. Each token represents a distinct pedestrian behavior, and different agents may share the same token.

To foster diversity amongst different agents, our goal is to assign different behavior tokens to different agents to exhibit distinct behaviors. In other words, for every state s, agents should perform different actions a depending on the assigned z. More formally, this idea can be formalised using information theory by maximizing the mutual information $I((S,A);Z)$, where (S,A) is the joint distribution of S and A. $Z \sim p(z)$, S, and A represent the random variables for behavior, state, and action respectively. Additionally, the diverse actions performed for different z should arise for every state instead of exploiting only certain states. For this, we minimize $I(S,Z)$ as a regularizer. In sum, we maximize

$$\begin{aligned}\mathscr{F}(\theta) &\triangleq I((S,A);Z) - I(S;Z) \\ &= (H[Z] - H[Z \mid S,A]) - (H[Z] - H[Z \mid S]) \\ &= -H[Z \mid S,A] + H[Z \mid S],\end{aligned} \tag{1}$$

where H is the Shannon entropy. The first term implies it is easy to infer the behavior z given any (s,a). This makes sense intuitively as it means the agents are distinguishable due to their diverse behaviors and not behaving in a homogeneous way. The second term implies that the agents' behavior should not be distinguishable exclusively given s. It is intractable to compute $p(z|s)$ and $p(z|(s,a))$ by integrating all states, actions, and skills. So we approximate the posteriors with learned discriminators $q_\phi(z|s)$ and $q_\psi(z|(s,a))$. We instead optimize the variational lower bound derived using Jensen's Inequality [29]:

$$\begin{aligned}\mathscr{F}(\theta) &= -H[Z \mid S,A] + H[Z \mid S] \\ &= \mathbb{E}_{z\sim p(z),s\sim\pi(z)}[\log p(z \mid s)] \\ &\quad - \mathbb{E}_{z\sim p(z),s\sim\pi(z),a\sim\pi(s,z)}[\log p(z \mid s,a)] \\ &\geq \mathbb{E}_{z\sim p(z),s\sim\pi(z)}[\log q_\phi(z \mid s)] \\ &\quad - \mathbb{E}_{z\sim p(z),s\sim\pi(z),a\sim\pi(s,z)}[\log q_\psi(z \mid s,a)] \\ &\triangleq \mathbb{G}(\theta),\end{aligned}$$

where $s \sim \pi(z)$ means to first sample the action a from π followed by sampling the environment to get the state s. It is non-trivial to directly optimize θ via maximizing the lower bound $\mathbb{G}(\theta)$ since $s \sim \pi(z)$ has to be sampled through a non-differentiable simulator. Below we introduce how to optimize θ using an intrinsic reward alongside the RL objective.

B. Behavior-Conditioned Policy

First, we incorporate the idea of behaviors into our policy where we condition our policy on the agent's behaviors. Each agent, i, sample their actions from a shared behavior-conditioned policy as $a \sim \pi_\theta(\cdot, s_t^i | z^i)$, for behavior token ID z^i at timestep t. Each behavior token maps to an embedding in the policy network, enabling the policy to generate distinct behaviors for agents. To maximize such diversity, we introduce an intrinsic pseudo-reward r_{int} motivated from maximizing $\mathbb{G}(\theta)$ derived previously:

$$r_t^{int} = log[q_{\psi_{sa}}(z \mid s_t, a_t)] - log[q_{\psi_s}(z \mid s_t)]. \tag{2}$$

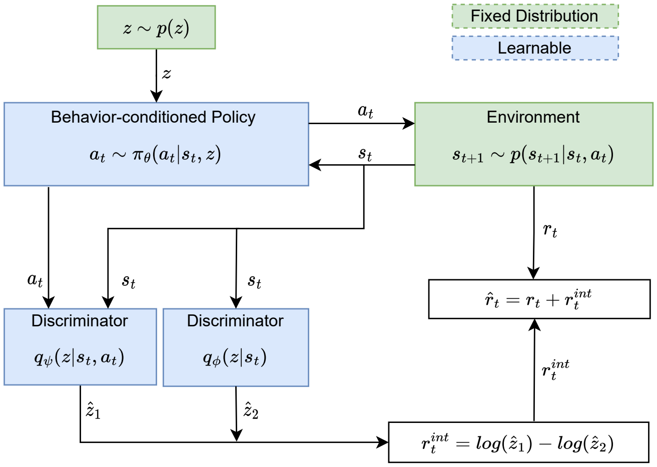

Fig. 2: **Our framework for behavior-conditioned policy**. An intrinsic reward is computed based on discriminators q_ψ and q_ϕ, which encourages the diversity by indirectly maximizing the lower variation bound $\mathbb{G}(\theta)$

Maximizing the intrinsic pseudo-reward through reinforcement learning allows maximizing $\mathbb{G}(\theta)$ despite sampling $s \sim \pi(z)$ from a non-differentiable simulator. In Eqn. (2), a_t is sampled from a policy conditioned on behavior rather than a default policy, as generating diverse actions requires knowledge about z. The proposed intrinsic reward promotes action diversity while learning the main task. Overall, our intrinsic reward shares some similarity to DIAYN [22] in which both use token-conditioned policies. However, the difference is that our method encourages agents with different tokens to generate diverse actions for a given state instead visiting diverse states. Figure 2 shows an overview of the interaction between the behavior-conditioned policy and the discriminators.

C. Training Procedure

We adapt the training procedure from [5], alternating between sampling trajectories and updating the policy via the PPO algorithm [31]. Each agent uses an identical policy to collect data until a batch is gathered. Algorithm 1 outlines the training details. Key differences from [5] are highlighted in blue: (1) At each episode start, agent i samples a new behavior token $z^i \sim p_M(z)$, with $p_M(z)$ being a discrete uniform distribution with M behaviors (Line 6). This token is mapped to a 32-dimensional continuous embedding. (2) Agents sample from a policy conditioned on z^i (Line 7), allowing for varied actions based on behavior. (3) Intrinsic rewards for each agent are computed using discriminators $q_{\psi_{sa}}$ and q_{ψ_s}, parameterized by ψ_{sa} and ψ_s respectively, based on Eqn. (2). These rewards are added to the task reward in the replay buffer (Line 8). (4) We optimize discriminators $q_{\psi_{sa}}$ and q_{ψ_s} with cross-entropy loss (Line 25) using the Adam optimizer [32]. Adding one standard deviation of Gaussian noise to discriminator inputs helps prevent overfitting. The loss is computed between predicted behavior tokens $\hat{z}$ from on-policy samples and ground truth behavior.

Algorithm 1 Behavior-conditioned policy for N agents

```
Initialize policy network $\pi_\theta$, value function $V_\phi$, discriminators $q_{\psi_{sa}}$ and $q_{\psi_s}$
Require: hyper-parameters $\alpha, \gamma, \lambda, \varepsilon, M$
while not converged do
    // Collect data in parallel
    for $i = 1, 2, \ldots, \mathrm{N}$ do
        Sample behavior $z^i \sim p_M(z)$
        Sample behavior-conditioned policy $\pi_\theta(\cdot, s_t|z)$ for $T_i$ timesteps, collecting $\{s^i_{t+1}, a^i_t, r^i_t\}$ where $t \in [0, T_i]$
        Modify reward by adding bonus intrinsic reward $r^i_t \leftarrow r^i_t + \alpha \left\{log[q_{\psi_{sa}}(z \mid (s^i_t, a^i_t)]) - log[q_{\psi_s}(z \mid s^i_t)]\right\}$
        Compute advantages using GAE [30] $\hat{A}^i_t = \sum_{l=0}^{T_i} (\gamma\lambda)^l (r^i_t + \gamma V_\phi\left(s^{t+1}_i\right) - V_\phi\left(s^i_t\right))$
    end for
    $\pi_{old} \leftarrow \pi_\theta$
    // Update Policy, Value Functions and Discriminators
    for $j = 1$ to $epoch_\pi$ do
        Compute Ratio $k_t = \frac{\pi_\theta\left(a^i_t|o^i_t\right)}{\pi_{\mathrm{old}}\left(a^i_t|o^i_t\right)}$
        $\mathbb{L}^{PPO\_clip}(\theta) = \sum_{t=1}^{T_{\max}} min\left(k_t\hat{A}^i_t, \mathrm{clip}\left(k_t, 1-\varepsilon, 1+\varepsilon\right)\hat{A}^i_t\right)$
        Update $\theta$ using Adam w.r.t. $\mathbb{L}^{PPO\_clip}(\theta)$
    end for
    for $j = 1$ to epoch_v do
        $\mathbb{L}^V(\phi) = -\sum_{i=1}^N \sum_{t=1}^{T_i} \left(\sum_{t'>t} \gamma^{t'-t} r^i_{t'} - V_\phi\left(s^i_t\right)\right)^2$
        Update $\phi$ using Adam w.r.t. $\mathbb{L}^V(\phi)$
    end for
    for $j = 1$ to epoch_d do
        $\mathbb{L}^D(\psi_{sa}) = -\sum_{i=1}^N \sum_{t=1}^{T_i} \left(z^i \cdot \log\left(q_{\psi_{sa}}(s^i_t, a^i_t)\right)\right)$
        $\mathbb{L}^D(\psi_s) = -\sum_{i=1}^N \sum_{t=1}^{T_i} \left(z^i \cdot \log\left(q_{\psi_s}(s^i_t)\right)\right)$
        Update $\psi_{sa}$, $\psi_s$ using Adam w.r.t. $\mathbb{L}^D(\psi_{sa})$, $\mathbb{L}^D(\psi_s)$
    end for
end while
```

IV. Experiments

We conduct comprehensive experiments to demonstrate the effectiveness of our method over previous solutions. We simulate these experiments with new crowd behaviors not encountered during agent training.

A. Experimental Setup

Implementation. We simulate a large-scale group of robots using Stage [33], a popular robot simulator widely used in multi-agent research. Each agent is initialized as a non-holonomic differential drive robot ($0.5m \times 0.5m$) equipped with a 2D-laser scanner to sense its surroundings. The 2D laser is set to 360 degrees FOV with a max range of 10m.

For agent states, rewards and neural network architecture, we follow the same setup as [5]. We make one change to the NN backbone by adding a behavior embedding derived from the behavior token, z, for the neural network input. Each discriminator is modeled with a two-layer feed-forward network with 128 hidden units and ReLU activations [34]. Table I lists the hyper-parameter settings.

Hyper-parameter	Value
Discount Factor γ	0.99
PPO Smoothing λ	0.95
PPO Clip Value ε	0.1
# Epoch for Policy Network	3
# Epoch For Value Network	3
# Epoch for Discriminators	1
Advantange Weight α	0.1
PPO Learning Rates	0.00005
Discriminators Learning Rates	0.00005
Epoch$_\pi$, Epoch$_d$, Epoch$_v$	3

TABLE I: **Hyper-parameters in our implementation.**

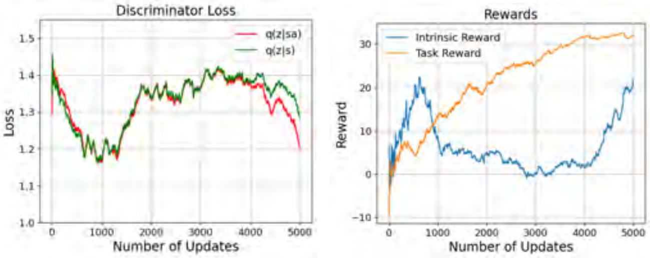

Fig. 3: The discriminator loss and reward curves.

Training Setup. We train agents in a realistic, heavily trafficked $20m \times 20m$ room. Random goals are placed at least $10m$ away from the agents' initial positions. A crash event is registered if laser-scan values fall below $0.5m$.

Testing Setup. To evaluate the robustness of our policy in unseen crowd behaviors, we propose a novel set of testing conditions with the criteria that the movements of the other agents must be unique and not encountered during training. Also, we omit static obstacles as we focus on dynamic obstacle avoidance. In these environments, our agent interacts with other agents which exhibit dynamic movements beyond our control. For clarity, we refer to other agents as pedestrians for the remainder of this section. In total, we design six different pedestrian setups to be evaluated for each study:

1) *Non-homogeneous* (NH): To achieve non-homogeneous behaviors, each pedestrian utilizes a distinct policy instance, initialized with a unique seed, to generate varied experiences during training.
2) *Invisible* (IN): Similar to 1), but our robot is invisible to pedestrians achieved by lowering our agent's height below the pedestrians' sensors. This reflects the non-reactive individuals in the real world.
3) *Variability* (VA): Similar to 2), pedestrians are invisible but receive random speed multipliers (0.5-1.5) at each episode start, representing real-world walking speed variability.
4) *Sub-optimal* (SO): Similar to 1), but the policy is trained for half the period (2.5k updates instead of 5k), simulating sub-optimal walking trajectories.
5) *Velocity-obstacle* (VO): We utilizes ground truth positions of all pedestrians to compute permissible velocities, following the method in [35].
6) *Social force* (SF): We use a force-based system to anticipate pedestrian movements following [36], and utilize ground positions for prediction similar to VO.

We consider a mixture of learning-based (NH,IN,VA,SO) and non-learning based (VO,SF) policies. Four of them (IN,SO,VO,SF) are challenging with the non-reactive pedestrians. All evaluations are repeated for 1000 episodes. If not stated explicitly, we set the number of robots $N = 5$ (1 agent and 4 pedestrians) and number of behaviors $M = 5$. Agents, pedestrians and goals are spawned similarly to training. During testing, each agent utilises a fixed behavior token, $z = 0$ for all episodes. Our primary metric is the 'success rate', without further classifying the non-successful episodes, as collisions are the primary reason instead of timeouts.

M	#Updates	#Updates/M	Pedestrian Type						Avg
			NH	IN	VA	SO	VO	SF	
1	5K	5K	0.63	0.55	0.59	0.52	0.59	0.43	0.55
5	5K	1K	**0.87**	**0.85**	**0.86**	**0.72**	**0.78**	**0.77**	**0.81**
10	5K	500	0.80	0.75	0.82	0.72	0.76	0.69	0.76
20	5K	250	0.61	0.63	0.54	0.59	0.59	0.51	0.58
10	10K	1K	0.88	0.86	0.86	0.76	0.77	0.78	0.82
20	20K	1K	0.89	0.87	0.85	0.77	0.78	0.79	0.82

TABLE II: **Impact of the number of behaviors** M. Policies are evaluated under six diverse unseen pedestrian setups.

N	Diversity	Pedestrian Type						Avg
		NH	IN	VA	SO	VO	SF	
5	No	0.63	0.55	0.59	0.52	0.59	0.43	0.55
5	Yes	**0.87**	**0.85**	**0.86**	**0.72**	**0.78**	**0.77**	**0.81**
10	No	0.41	0.38	0.42	0.37	0.31	0.28	0.35
10	Yes	0.83	0.78	0.77	0.65	0.75	0.52	0.72
20	No	0.47	0.43	0.42	0.39	0.36	0.32	0.38
20	Yes	0.58	0.50	0.58	0.47	0.42	0.49	0.50

TABLE III: **Impact of the number of agents** N**.**

Intrinsic Reward	α_{best}	Pedestrian Type						Avg	$\mathscr{D}$
		NH	IN	VA	SO	VO	SF		
$log[q_1(z \mid s,a)] - log[q_2(z \mid s)]$	0.10	**0.87**	**0.85**	**0.86**	**0.72**	**0.78**	**0.77**	**0.81**	1.10
$log[q(z \mid s,a)]$	0.01	0.88	0.77	0.86	0.70	0.75	0.72	0.78	0.48
$log[q(z \mid s)]$ [22]	0.03	0.75	0.75	0.69	0.71	0.73	0.58	0.70	0.17
None	0	0.63	0.55	0.59	0.52	0.59	0.43	0.55	0

TABLE IV: **Policies trained using different intrinsic rewards**. $\mathscr{D}$ is a measure of action diversity between agents

B. Training Results

Figure 3 shows the stability of the discriminators and the intrinsic reward during training. The discriminator loss and intrinsic reward steadily improves until ~ 700 updates, which then flattens until ~ 4000 updates, possibly due to novel state-action exploration. Subsequently, both curves continue to improve again until the end of training where the main task has converged. The best advantage weight α, which combines the task and intrinsic reward is 0.1.

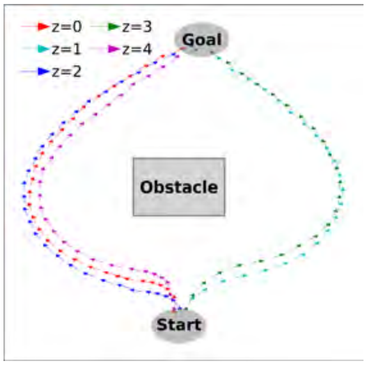

Fig. 4: Agents' varied paths based on sampled behaviors.

Next, we investigate if behavior-conditioned policies exhibit different behaviors to reach the desired goal. We record the behaviors of agents starting from the origin and reaching a fixed goal behind a static obstacle. The trajectories of the 5 agents can be seen in Figure 4. These agents exhibit different behaviors when conditioned on different tokens z. While passing the obstacle, some agents are left-inclined whereas some are right-inclined which is consistent with the motivation illustrated in Figure 1. Additionally, we also observe the velocity diversity which is represented by the length of the arrow. For example, the blue agent ($z = 2$) travels slightly faster than the purple agent ($z = 4$) when taking a slightly longer route. While the diversity may appear slight, deploying it with multiple agents generates vastly different training trajectories, enhancing agent robustness. More qualitative examples about the diversity among dynamic obstacles are available on https://youtu.be/EevMn2-ZNng.

C. Impact of the Number of Behaviors

We assess diversity's impact on agent robustness and its scalability with behavior count M. When $M = 1$, all agents share the same behavior, equivalent to the baseline policy in [5]. Our results are presented in Table II. We have the following three observations. First, having non-reactive pedestrians (NH,IN,VA,SO) is generally a more difficult task with fewer successful runs. Second, adding diversity ($M \neq 1$) outperforms the default policy ($M = 1$) for all pedestrian types, including the challenging non-reactive pedestrians. This validates the effectiveness of our proposed method. Third, the optimal number of behaviors is $M = 5$ and the effect of diversity starts to diminish as we scale to higher values of $M = 10$ and 20. We hypothesize the diminishing effect is resulted from less frequent sampling when M increases. To investigate this, we increase $num\_updates/M$ for $M = 10$ and 20 to match $M = 5$ and find that the performance of $M = 10$ and $M = 20$ could match that of $M = 5$, validating our hypothesis. However, this comes at an expense of more updates. Overall, $M = 5$ provides a good balance between creating good diversity and sample efficiency.

D. Scalability

Here, we investigate the effect of increased numbers of agents N and report the results in Table III. As the number of agents, N, increases to 10 and 20, the scenes become more crowded, making it harder for them to reach their goals. This negatively affects the convergence speed, as agents get restarted more frequently due to collisions or other obstacles. Despite this, by adding diversity to the policy, we achieve consistent performance improvements across all pedestrian cases.

E. Intrinsic Rewards

In Section III-A, we formulate a cost function to promote diversity among agents. In the formulation, we require that diverse actions performed for different z should arise for every state instead of exploiting only certain states. This is achieved using a regularizer as part of the intrinsic reward proposed in Eqn. (2). From ablation experiments reported in Table IV, we observe that the policy trained with the intrinsic reward containing the regularization term, $-log(q \mid s)$, outperforms the policy without this term. The performance improvement is consistent in all pedestrian setups including the challenging non-reactive pedestrians. Despite this, the policy trained without regularization still outperforms the base policy without the intrinsic reward.

Next, we compare our method with state-space exploration based intrinsic rewards, DIYAN, from [22] which may implicitly encourage action diversity through novel state exploration. However, it still lacks action diversity compared to our proposed intrinsic reward in Eqn.(2), where the diversity of the action is explicitly encouraged. To measure the action

Metrics	Policy	Pedestrian Type					
		NH	**IN**	**VA**	**SO**	**VO**	**SF**
Success Rate ↑	Basic	0.63	0.55	0.59	0.52	0.59	0.43
	Safe	0.86	0.77	0.81	0.67	0.69	0.61
	Ours	**0.87**	**0.85**	**0.86**	**0.72**	**0.78**	**0.77**
Extra Time (s) ↓	Basic	2.833 ± 2.439	3.366 ± 2.621	3.724 ± 2.219	2.511 ± 1.751	3.158 ± 2.121	2.997 ± 1.136
	Safe	5.041 ± 2.356	5.102 ± 2.719	5.248 ± 2.658	5.100 ± 2.335	5.217 ± 2.454	4.813 ± 2.348
	Ours	**2.712 ± 2.259**	**2.902 ± 2.671**	**2.714 ± 2.427**	**2.336 ± 0.995**	**2.119 ± 1.038**	**2.202 ± 1.344**
Extra Distance (m) ↓	Basic	4.811 ± 4.011	4.123 ± 5.637	5.717 ± 3.013	4.197 ± 5.873	5.887 ± 4.187	3.321 ± 3.899
	Safe	10.099 ± 4.452	10.734 ± 5.177	16.601 ± 5.235	10.116 ± 4.182	9.870 ± 3.946	10.024 ± 4.275
	Ours	**3.667 ± 3.587**	**3.930 ± 4.024**	**4.262 ± 4.489**	**3.217 ± 2.309**	**2.492 ± 1.719**	**2.662 ± 2.112**
Average Speed (m/s) ↑	Basic	0.919 ± 0.096	0.927 ± 0.088	0.917 ± 0.87	0.922 ± 0.068	0.920 ± 0.079	0.910 ± 0.087
	Safe	0.810 ± 0.091	0.795 ± 0.098	0.779 ± 0.100	0.811 ± 0.084	0.811 ± 0.084	0.786 ± 0.097
	Ours	**0.957 ± 0.059**	**0.955 ± 0.061**	**0.942 ± 0.076**	**0.960 ± 0.057**	**0.966 ± 0.039**	**0.965 ± 0.043**

TABLE V: **Comparisons with baseline methods using different metrics averaged across 1000 episodes.**

diversity, we also introduce a new metric, $\mathscr{D}$, using the KL divergence of action distributions between pairwise agents:

$$\mathscr{D} = \frac{1}{|\tau| N_{i \neq j}} \sum_{s \in \tau} \sum_{i \neq j} \mathrm{KL}\left(\pi(a|s, z=i) \| \pi(a|s, z=j)\right)$$

where τ denotes a trajectory. Specifically, we collect a trajectory of 1000 steps using the trained policy with no intrinsic reward. From Table IV, our proposed method achieves higher action diversity $\mathscr{D}$ than the state-space exploration based intrinsic rewards. Also, we observe that higher values of $\mathscr{D}$ get translated into higher robustness in unseen crowd behaviors, achieving a greater success rate.

F. Comparisons with Prior Work

We quantitatively compare our proposed behavior-conditioned policy with existing solutions to demonstrate its robustness. In particular, we set up the baseline method as described in [5], equivalent to our proposed method with $M = 1$. Additionally, we added a safe policy proposed in [10], which uses safety zone rewards to encourage safe behaviors, which could crash less in unseen crowd movements. For our proposed method, we utilize the model trained with $M = 5$ and $N = 5$. Table V shows the comparison results against different metrics across 1000 episodes. Each metrics (success rate, extra time, extra distance, average speed) are similarly defined like in [5].

Our proposed method consistently outperforms others across various pedestrian types, suggesting robust strategies for handling diverse crowd behaviors effectively. While the safe policy achieves a higher success rate than the base policy, it slightly falls short of our proposed policy. Collisions primarily contribute to non-successful episodes, surpassing timeouts. The safe policy with a safety buffer performs well in reactive setups (NH, VA), closely matching our policy's results. However, it struggles in non-reactive setups (IN, SO, VO, SF). The conservative behavior of the safe policy reduces collisions but increases time and distance compared to our proposed policy, sacrificing other metrics. Specifically, both time and distance taken by the safe policy are more than double those of our proposed policy.

G. Realistic Deployment

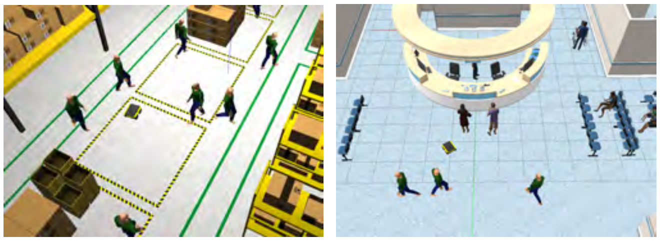

Fig. 5: **Testing our method in Gazebo with more realistic scenarios**. Map settings: (Left) Warehouse (Right) Hospital

To validate our method in a more realistic setup, we deploy our best performing policy ($N = 5$, $M = 5$) on a Jackal robot in Gazebo simulator [37]. We utilize two maps (warehouse and hospital) from Arena-ROSNAV-3D [38] as seen in Figure. 5. For each episode, we randomly assign the start and goal position with 3 to 8 pedestrians within an open area of each map ($\sim 10m \times 10m$). The pedestrians movements are simulated using the social force model [36].

Table VI shows the success rate from 100 episodes with and without the diversity consideration. Our method proves to be equally effective for realistic scenes, outperforming the baseline method when diversity is used during training. More qualitative examples for realistic scenes are available on https://youtu.be/EevMn2-ZNng.

Diversity	Warehouse	Hospital
No	0.40	0.37
Yes	**0.81**	**0.69**

TABLE VI: **Success rate out of 100 episodes**

V. Conclusion

This paper introduces a framework to increase an agent's ability to generalize to unseen crowd behaviors by utilizing diverse behaviors in a sample-efficient manner. Adding diversity in a multi-agent framework implicitly provides each agent with a more varied range of experiences, hence increasing its generalizability of unseen crowd behaviors. We demonstrate the robustness of the proposed method in an extensive set of evaluation scenes containing challenging pedestrians' behaviors. We also validate the scalability of our solution and practicality in realistic scenes. Our experiments also demonstrate that our method improves the success rates without negatively affecting other important metrics.

Acknowledgment

This study is supported under RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

[1] M. Ben-Ari, F. Mondada, M. Ben-Ari, and F. Mondada, "Robots and their applications," *Elements of robotics*, pp. 1–20, 2018.

[2] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 31–36.

[3] 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.

[4] 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.

[5] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 6252–6259.

[6] M. Pfeiffer, S. Shukla, M. Turchetta, C. Cadena, A. Krause, R. Siegwart, and J. Nieto, "Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4423–4430, 2018.

[7] T. Fan, P. Long, W. Liu, J. Pan, R. Yang, and D. Manocha, "Learning resilient behaviors for navigation under uncertainty," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5299–5305.

[8] T. Fan, X. Cheng, J. Pan, P. Long, W. Liu, R. Yang, and D. Manocha, "Getting robots unfrozen and unlost in dense pedestrian crowds," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1178–1185, 2019.

[9] J. Liang, U. Patel, A. J. Sathyamoorthy, and D. Manocha, "Crowd-steer: Realtime smooth and collision-free robot navigation in densely crowded scenarios trained using high-fidelity simulation," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 4221–4228.

[10] J. Jin, N. M. Nguyen, N. Sakib, D. Graves, H. Yao, and M. Jagersand, "Mapless navigation among dynamics with social-safety-awareness: a reinforcement learning approach from 2d laser scans," in *IEEE international conference on robotics and automation*, 2020.

[11] Y. Zhou, S. Li, and J. Garcke, "R-sarl: Crowd-aware navigation based deep reinforcement learning for nonholonomic robot in complex environments," *arXiv preprint arXiv:2105.13409*, 2021.

[12] B. Brito, M. Everett, J. P. How, and J. Alonso-Mora, "Where to go next: Learning a subgoal recommendation policy for navigation in dynamic environments," *IEEE Robotics and Automation Letters*, 2021.

[13] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[14] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The international journal of robotics research*, vol. 17, no. 7, pp. 760–772, 1998.

[15] Y. Wang, H. He, and C. Sun, "Learning to navigate through complex dynamic environment with modular deep reinforcement learning," *IEEE Transactions on Games*, vol. 10, no. 4, pp. 400–412, 2018.

[16] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *International Conference on Intelligent Robots and Systems*, 2018.

[17] Q. Tan, T. Fan, J. Pan, and D. Manocha, "Deepmnavigate: Deep reinforced multi-robot navigation unifying local & global collision avoidance," in *International Conference on Intelligent Robots and Systems*, 2020.

[18] S. H. Semnani, H. Liu, M. Everett, A. De Ruiter, and J. P. How, "Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3221–3226, 2020.

[19] Y. Cui, H. Zhang, Y. Wang, and R. Xiong, "Learning world transition model for socially aware robot navigation," in *IEEE International Conference on Robotics and Automation*, 2021.

[20] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *The International Journal of Robotics Research*, 2020.

[21] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *International conference on machine learning*. PMLR, 2017, pp. 1352–1361.

[22] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, "Diversity is all you need: Learning skills without a reward function," in *International Conference on Learning Representations*, 2019.

[23] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *NeurIPS*, vol. 30, 2017.

[24] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," *Advances in neural information processing systems*, vol. 29, 2016.

[25] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 7234–7284, 2020.

[26] K. R. McKee, I. Gemp, B. McWilliams, E. A. Duèñez Guzmán, E. Hughes, and J. Z. Leibo, "Social diversity and social preferences in mixed-motive reinforcement learning," ser. AAMAS '20, 2020.

[27] C. Li, T. Wang, C. Wu, Q. Zhao, J. Yang, and C. Zhang, "Celebrating diversity in shared multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3991–4002, 2021.

[28] Y. Lee, J. Yang, and J. J. Lim, "Learning to coordinate manipulation skills via skill behavior diversification," in *International conference on learning representations*, 2020.

[29] D. Barber and F. Agakov, "The im algorithm: a variational approach to information maximization," *NeurIPS*, 2004.

[30] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] R. Vaughan, "Massively multi-robot simulation in stage," *Swarm intelligence*, vol. 2, pp. 189–208, 2008.

[34] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[35] 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.

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

[37] 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.

[38] L. Kästner, T. Bhuiyan, T. A. Le, E. Treis, J. Cox, B. Meinardus, J. Kmiecik, R. Carstens, D. Pichel, B. Fatloun, N. Khorsandi, and J. Lambrecht, "Arena-bench: A benchmarking suite for obstacle avoidance approaches in highly dynamic environments," 2022. [Online]. Available: https://arxiv.org/abs/2206.05728