Magnetic Field-Based Reward Shaping for Goal-Conditioned Reinforcement Learning

Hongyu Ding, Yuanze Tang, Qing Wu, Bo Wang, *Member, IEEE*, Chunlin Chen, *Senior Member, IEEE*, and Zhi Wang, *Member, IEEE*

***Abstract*—Goal-conditioned reinforcement learning (RL) is an interesting extension of the traditional RL framework, where the dynamic environment and reward sparsity can cause conventional learning algorithms to fail. Reward shaping is a practical approach to improving sample efficiency by embedding human domain knowledge into the learning process. Existing reward shaping methods for goal-conditioned RL are typically built on distance metrics with a linear and isotropic distribution, which may fail to provide sufficient information about the ever-changing environment with high complexity. This paper proposes a novel magnetic field-based reward shaping (MFRS) method for goal-conditioned RL tasks with dynamic target and obstacles. Inspired by the physical properties of magnets, we consider the target and obstacles as permanent magnets and establish the reward function according to the intensity values of the magnetic field generated by these magnets. The nonlinear and anisotropic distribution of the magnetic field intensity can provide more accessible and conducive information about the optimization landscape, thus introducing a more sophisticated magnetic reward compared to the distance-based setting. Further, we transform our magnetic reward to the form of potential-based reward shaping by learning a secondary potential function concurrently to ensure the optimal policy invariance of our method. Experiments results in both simulated and real-world robotic manipulation tasks demonstrate that MFRS outperforms relevant existing methods and effectively improves the sample efficiency of RL algorithms in goal-conditioned tasks with various dynamics of the target and obstacles.**

***Index Terms*—Dynamic environments, goal-conditioned reinforcement learning, magnetic field, reward shaping.**

I. Introduction

REINFORCEMENT learning (RL) [1] is a general optimization framework of how an autonomous active agent learns an optimal behavior policy that maximizes the cumulative reward while interacting with its environment in a trial-and-error manner. Traditional RL algorithms, such as dynamic programming [2], Monte Carlo methods [3], and temporal difference learning [4], have been widely applied to Markov decision processes (MDPs) with discrete state and action space [5]–[8]. In recent years, with advances in deep learning, RL combined with neural networks has led to great success in high-dimensional applications with continuous space [9]–[13], such as video games [14]–[16], the game of Go [17], robotics [18], [19], and autonomous driving [20].

While RL has achieved significant success in various benchmark domains, its deployment in real-world applications is still limited since the environment is usually dynamic, which can cause conventional learning algorithms to fail. Goal-conditioned reinforcement learning [21]–[23] is a typical dynamic environment setting where the agent is required to reach ever-changing goals that vary for each episode by learning a general goal-conditioned policy. The universal value function approximator (UVFA) [22] makes it possible by letting the Q-function depend on not only a state-action pair but also a goal. In this paper, we treat the obstacle as an opposite type of the goal in goal-conditioned RL, i.e., “anti-goal” that the agent should avoid. Following UVFA [22], we can naturally give a positive reward when the agent reaches the target, a negative one when hitting on the obstacles, and 0 otherwise. However, the resulting sparse rewards make it difficult for the agent to obtain valid information and lead to poor sample efficiency [24], [25]. Especially in the dynamic environment, the problem is more severe since the inherently sparse reward is constantly changing over time [23].

Reward shaping [26]–[29] effectively addresses the sample efficiency problem in sparse reward tasks without changing the original optimal policy. It adds an extra shaping reward to the original environment reward to form a new shaped reward that the agent applies to update policy, offering additional dense information about the environment with human domain knowledge. Potential-based reward shaping (PBRS) [26] defines the strict form of the shaping reward function as the difference between the potential functions of the successor and the current state. It ensures that the optimal policy learned from the shaped reward is consistent with the one learned from the original reward, also known as the optimal policy invariance property.

The shaping reward in existing methods is typically set

Manuscript received November 18, 2022; revised December 30, 2022; accepted January 29, 2023. This work was supported in part by the National Natural Science Foundation of China (62006111, 62073160) and the Natural Science Foundation of Jiangsu Province of China (BK20200330). Recommended by Associate Editor Nian Zhang. *(Corresponding author: Zhi Wang.)*

Citation: H. Y. Ding, Y. Z. Tang, Q. Wu, B. Wang, C. L. Chen, and Z. Wang, “Magnetic field-based reward shaping for goal-conditioned reinforcement learning,” *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 12, pp. 1–15, Dec. 2023.

H. Y. Ding, B. Wang, C. L. Chen, and Z. Wang are with the Department of Control Science and Intelligence Engineering, School of Management and Engineering, Nanjing University, Nanjing 210093, China (e-mail: hongyuding@smail.nju.edu.cn; bowangsme@nju.edu.cn; clchen@nju.edu.cn; zhiwang@nju.edu.cn).

Y. Z. Tang and Q. Wu are with the Department of Power Engineering and Process Machinery, School of Mechanical and Power Engineering, East China University of Science and Technology, Shanghai 200237, China (e-mail: yztang@mail.ecust.edu.cn; qwu@ecust.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.123477

according to a distance metric for goal-conditioned RL, e.g., the Euclidean distance between the agent and goal [30]. However, due to the linear and isotropic properties of the distance metrics, it may fail to provide adequate information about a dynamic environment with high complexity. Specifically, the derivative of the linear distance-based shaping reward is a constant, which cannot provide higher-order information on the optimization landscape. In this case, the agent only knows "where" to go, rather than "how" to go. On the other hand, the isotropic property also exacerbates this problem. Considering a 2D plain with an initial state and a target, the points with the same Euclidean distance from the target will share a same value of potential or shaping reward under distance-based setting. However, the points between the initial state and target should be given higher values since the optimal policy is supposed to be the steepest path from the initial state to the target. Consequently, it is necessary to develop a more sophisticated reward shaping method that can provide sufficient and conducive information for goal-conditioned tasks.

Intuitively, the agent should be attracted to the target and repelled by the obstacles, which is similar to the physical phenomenon that magnets of the same polarity repel each other while those of different polarities attract each other. Inspired by this, we propose a magnetic field-based reward shaping (MFRS) method for goal-conditioned RL tasks. We consider the dynamic target and obstacles as permanent magnets, and build the reward function according to the intensity values of the magnetic field generated by these magnets. The nonlinear and anisotropic properties of the generated magnetic field provide a sophisticated reward function that carries more accessible and sufficient information about the optimization landscape than in the distance-based setting. Besides, we use several normalization techniques to unify the calculated values of magnetic field intensity and prevent the value exploding problem. Then, we define the form of our magnetic reward function and transform it into a potential-based one by learning a secondary potential function concurrently to ensure the optimal policy invariance of our method.

In summary, our main contributions are as follows.

1) We propose a novel MFRS method for goal-conditioned RL tasks with dynamic target and obstacles.

2) We build a 3-D simulated robotic manipulation platform and verify the superiority of MFRS on a set of tasks with various goal dynamics.

3) We apply MFRS to a physical robot in the real-world environment and demonstrate the effectiveness of our method.

The rest of this paper is organized as follows. Section II introduces the preliminaries of goal-conditioned RL and related work. In Section III, we first present the overview of MFRS, followed by specific implementations in detail and the final integrated algorithm. Simulated experiments on several robotic manipulation tasks with the dynamic target and obstacles are conducted in Section IV. Section V demonstrates the application of our method to a real robot, and Section VI presents our concluding remarks.

II. Preliminaries and Related Work

A. Goal-Conditioned Reinforcement Learning

We consider a discounted infinite-horizon goal-conditioned Markov decision process (MDP) framework, defined by a tuple $\langle \mathcal{S}, \mathcal{G}, \mathcal{A}, \mathcal{T}, R, \gamma \rangle$, where $\mathcal{S}$ is the set of states, $\mathcal{G}$ is the set of goals which is the subset of states $\mathcal{G} \subseteq \mathcal{S}$, $\mathcal{A}$ is the set of actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ is the state transition probability, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \rightarrow \mathbb{R}$ is the goal-conditioned reward function, and γ is a discount factor $\gamma \in (0,1]$.

We aim to learn a goal-conditioned policy that can achieve multiple goals simultaneously. Considering a goal-reaching agent interacting with the environment, in which every episode starts with sampling an initial state $s_0 \sim \rho_0$ and a goal $g \sim \rho_g$, where ρ_0 and ρ_g denote the distribution of the initial state and goal, respectively. At every timestep t of the episode, the agent selects an action according to a goal-conditioned policy $a_t \sim \pi(\cdot|s_t, g)$. Then it receives an instant reward $r_t = r(s_t, a_t, g)$ that indicates whether the agent has reached the goal, and the next state is sampled from the distribution of state transition probability $s_{t+1} \sim p(\cdot|s_t, a_t)$.

Formally, the objective of goal-conditioned RL is to find an optimal policy π^* that maximizes the expected return, which is defined as the discounted cumulative rewards

$$J(\pi) = \mathbb{E}_{g \sim \rho_g, \tau \sim d^{\pi}(\cdot|g)} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right] \tag{1}$$

under the distribution

$$d^{\pi}(\tau|g) = \rho_0(s_0) \prod_{t=0}^{\infty} \pi(a_t|s_t, g) p(s_{t+1}|s_t, a_t) \tag{2}$$

where $\tau = (s_0, a_0, s_1, a_1, ...)$ is the learning episode. Based on UVFA [22], goal-conditioned RL algorithms rely on the appropriate estimation of goal-conditioned action-value function $Q^{\pi}(s, a, g)$, which describes the expected return when performing action a in state s, goal g and following π after:

$$Q^{\pi}(s_t, a_t, g) = \mathbb{E}_{(r_{i \geq t}, s_{i > t}) \sim E, (a_{i > t}) \sim \pi, g \sim \rho_g} \left[\sum_{i=t}^{\infty} \gamma^{i-t} r_i \right] \tag{3}$$

where E stands for the environment.

In this work, we follow the standard off-policy actor-critic framework [31]–[35], where a replay buffer D is used to store the transition tuples $(s_t, a_t, s_{t+1}, r_t, g)$ as the experience for training. The policy is referred as the actor and the action-value function as the critic, which can be represented as parameterized approximations using deep neural network (DNN) $\pi_{\boldsymbol{\theta}}(s, g)$ and $Q_{\boldsymbol{\phi}}(s, a, g)$, where $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ denote the parameters of the actor-network and critic-network, respectively.

Actor-critic algorithms maximize the expected return by alternating between policy evaluation and policy improvement. In the policy evaluation phase, the critic estimates the action-value function of the current policy $\pi_{\boldsymbol{\theta}}$, and the objective for the critic is minimizing the square of the Bellman error [2]

$$L(\boldsymbol{\phi}) = \mathbb{E}_{(s_t, a_t, s_{t+1}, r_t, g) \sim D} \left[y_t - Q_{\boldsymbol{\phi}}(s_t, a_t, g) \right]^2 \tag{4}$$

where $y_t = r_t + \gamma \mathbb{E}_{a_{t+1}\sim\pi_{\theta}(\cdot|s_t,g)}\left[Q_{\phi}(s_{t+1}, a_{t+1}, g)\right]$. In the policy improvement phase, the actor is optimized by maximizing the expected action-value function, and thus the objective function can be written as

$$J(\theta) = \mathbb{E}_{(s_t,g)\sim D, a_t\sim\pi_{\theta}(\cdot|s_t,g)}\left[Q_{\phi}(s_t, a_t, g)\right]. \tag{5}$$

Hereafter, a gradient descent step $\phi \leftarrow \phi - \beta\nabla_{\phi}L(\phi)$ and a gradient ascent step $\theta \leftarrow \theta + \alpha\nabla_{\theta}J(\theta)$ can be taken to update the parameters of the critic network and actor network, respectively, where β and α denote the learning rates. The minimizing and maximizing strategies continue until θ and ϕ converge [18], [36].

B. Related Work

Goal-conditioned RL aims to learn a universal policy that can master reaching multiple goals simultaneously [37]. Following UVFA [22], the reward function in goal-conditioned RL is typically a simple unshaped sparse reward, indicating whether the agent has reached the goal. Therefore, the agent can gain nothing from the environment until eventually getting into the goal region, which usually requires a large number of sampled episodes and brings about the problem of sample inefficiency, especially in complex environments such as robotic manipulation tasks.

A straightforward way to alleviate the sample efficiency problem is to replace the reward with a distance measure between the agent's current state and goal. References [38]–[42] propose different distance metrics to provide the agent with an accurate and informative reward signal indicating the distance to the target. However, they cannot guarantee that the optimal learned policy with the shaped reward will be the same as the original reward, which yields an additional local optima problem [30]. In contrast, our method holds the property of optimal policy invariance and preserves a fully-informative dense reward simultaneously.

For the local optima problem generated by directly replacing the reward function, reward shaping is a popular way of incorporating domain knowledge into policy learning without changing the optimal policy. Ng *et al.* [26] propose the potential-based reward shaping (PBRS), which defines the strict form of shaping reward function with proof of sufficiency and necessity that the optimal policy remains unchanged. Wiewiora *et al.* [27] extend the input of potential function to state-action pair, allowing the incorporation of behavioral knowledge. Devlin and Kudenko [28] introduce a time parameter to the potential function, allowing the generalization to dynamic potentials. Further, Harutyunyan *et al.* [29] combine the above two extensions and propose the dynamic potential-based advice (DPBA) method, which can translate arbitrary reward function into the form of potential-based shaping reward by learning a secondary potential function together with the policy. All these methods hold the theoretical property of optimal policy invariance. However, they typically calculate the shaping reward based on simple distance metrics in goal-conditioned RL, which may fail to provide sufficient information in a highly complex dynamic environment. In contrast, we establish our shaping reward function according to the magnetic field intensity with nonlinear and anisotropic distribution, carrying more informative and conducive information about the optimization landscape.

For the local optima problem, apart from the reward shaping methods, learning from demonstration (LfD) is also considered as an effective solution that introduces a human-in-the-loop learning paradigm with expert guidance. Reference [43] proposes a human guidance-based PER mechanism to improve the efficiency and performance of RL algorithm, and an incremental learning-based behavior model that imitates human demonstration to relieve the workload of human participants. Reference [44] develops a real-time human-guidance-based DRL method for autonomous driving, in which an improved actor-critic architecture with modified policy and value network is introduced. On the other hand, the local optima problem also exists when performing gradient descent to solve the ideal parameters in RL, especially in the field of control systems. To obtain the global optimal solution, [45] exploits the current and some of the past gradients to update the weight vectors in neural networks, and proposes a multi-gradient recursive (MGR) RL scheme, which can eliminate the local optima problem and guarantee a faster convergence than gradient descent methods. Reference [46] extends MGR to the distributed RL to deal with the tracking control problem of uncertain nonlinear multi-agent systems, which further decreases the dependence on network initialization.

Another line of research that tackles the sample efficiency problem due to sparse reward in goal-conditioned RL is the hindsight relabeling strategy, which can be traced back to the famous hindsight experience replay (HER) [23]. HER makes it possible to reuse unsuccessful trajectories in the replay buffer by relabeling the "desired goals" with certain "achieved goals", significantly reducing the number of sampled transitions required for learning to complete the tasks. Recently, researchers have been interested in designing advanced relabeling methods for goal sampling to improve the performance of HER [47]–[52]. However, HER and its variants only focus on achieving the desired goal but fail to consider the obstacles. In contrast, our method can handle tasks with multiple dynamic obstacles and learn to reach the desired goal while avoiding the obstacles with high effectiveness and efficiency.

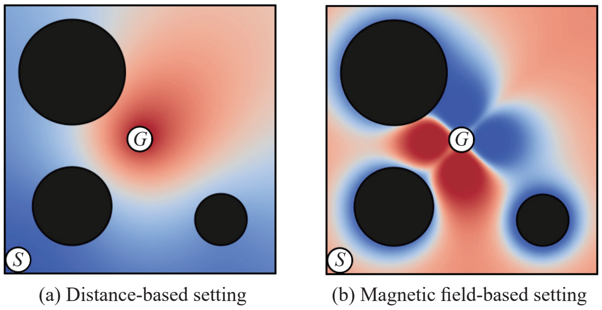
III. Magnetic Field-Based Reward Shaping

In this section, we first introduce the overview of MFRS that considers the target and obstacles as permanent magnets and exploits the physical property of the magnetic field to build the shaping reward function for goal-conditioned tasks. Then, we illustrate the calculation of the resulting magnetic field intensity for different shapes of permanent magnets in detail and use several normalization techniques to unify the intensity distribution and prevent the value exploding problem. Finally, we define the form of our magnetic reward function and transform it into a potential-based one by learning a secondary potential function concurrently to ensure the optimal policy invariance property.

A. Method Overview

For the sample inefficiency problem due to the sparse reward in goal-conditioned RL, reward shaping is a practical approach to incorporating domain knowledge into the learning process without changing the original optimal policy. However, existing reward shaping methods typically use distance measures with a linear and isotropic distribution to build the shaping reward, which may fail to provide sufficient information about the complex environments with the dynamic target and obstacles.

Therefore, it is necessary to develop a more sophisticated reward shaping method for this specific kind of task. For a goal-conditioned RL task with the ever-changing target and obstacles, our focus is to head toward the target while keeping away from the obstacles. Naturally, the agent is attracted by the target and repelled by the obstacles, which is analogous to the physical phenomenon that magnets of the same polarity repel each other while those of different polarities attract each other. Motivated by this, we model the dynamic target and obstacles as permanent magnets and establish our shaping reward according to the intensity of the magnetic field generated by these permanent magnets. The magnetic field with a nonlinear and anisotropic distribution can provide more accessible and conducive information about the optimization landscape, thus introducing a more sophisticated shaping reward for goal-conditioned RL tasks with dynamic target and obstacles. Fig. 1 visualizes an example comparison of the shaping reward distribution between the traditional distance-based setting and our magnetic field-based setting.



(a) Distance-based setting (b) Magnetic field-based setting

Fig. 1. An example comparison of the shaping reward distribution between the traditional distance-based setting and our magnetic field-based setting. *S* and *G* are the start point and goal. Black circles indicate the obstacles in different sizes. Since the shaping reward is positively related to the magnetic field intensity, areas painted with warmer color possess a larger shaping reward.

According to the physical property of permanent magnets, the agent will sense a higher intensity value in the magnetic field when getting closer to the target or away from the obstacles. The shaping reward varies more intensively in the near region of the target and obstacles. The target area uses a far larger shaping reward to facilitate its attraction to the agent, while the area around obstacles will receive a far smaller shaping reward as a “danger zone” to keep the agent at a distance. The nonlinearity of the magnetic field reinforces the process that the agent is attracted by the target and repelled by the obstacles. Being different from the distance-based setting, the derivative of the magnetic field intensity value with respect to the agent’s position will also increase when approaching the target or moving away from the obstacles. This property makes the shaping reward vary more intensively when the agent explores the environments, especially in the near regions around the target and obstacles, as illustrated in Fig. 1(b). Under this setting, the agent can be informed of an extra variation trend of the reward function together with the reward value itself. Thus it knows “how” to approach the target in addition to “where” to go. Therefore, the nonlinear characteristic of the magnetic field can provide more useful position information for the algorithm to improve its performance.

For a goal-conditioned RL task, the optimal policy should be the steepest path from the initial state to the target while avoiding the obstacles. Considering the points with the same Euclidean distance to the target, the distance-based reward shaping methods assign the same shaping reward (or potential) value to these points according to the isotropic distribution of the distance function. However, the points between the initial state and the target should be preferred than the others since they can eventually lead to a shorter path to the target. It is intuitive and rational to assign higher values to those points, which can more explicitly embed the orientation information of the target into the shaping reward. Our magnetic field-based approach can realize this consideration by adjusting the target magnet’s magnetization direction towards the agent’s initial state. As shown in Fig. 1(b), the shaping reward distribution of the target area is anisotropic, where the part between the initial state and the goal exhibits a larger value than the other parts. Accordingly, the agent will have a clear insight of the heading direction towards the target. In brief, the anisotropic feature of the magnetic field can provide more orientation information about the goal-conditioned RL tasks.

Based on the above insights, we propose a MFRS method for goal-conditioned RL tasks, which can provide abundant and conducive information on the position and orientation of the target and obstacle due to the nonlinear and anisotropic distribution of the magnetic field. To this end, an additional dense shaping reward according to the magnetic field intensity is added to the original sparse reward with the guarantee of optimal policy invariance. As a result, the agent can enjoy an informative reward signal indicating where and how far the target and obstacles are in each timestep, and maintain the same optimal policy under the sparse reward condition. Therefore, the sample inefficiency problem caused by the sparse reward can be alleviated. We consider the target and obstacles as permanent magnets, and calculate the intensity values of the resulting magnetic field using physical laws.

B. Calculation of Magnetic Field Intensity

For permanent magnets in regular shapes, we can directly calculate the intensity function analytically, using different formulas for different shapes. We take the spherical and cuboid permanent magnets as examples to calculate the distribution of magnetic field intensity in three-dimensional (3-D) space for illustration. As for the permanent magnets in irregular shapes, the intensity distribution of magnetic field can be obtained by analog platforms using physics simulation software such as COMSOL [53].

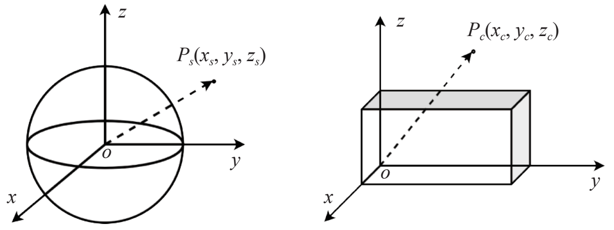
We assume the spherical and cuboid magnets are both saturatedly magnetized along the positive direction of the z-axis, and Fig. 2 presents the magnetic field coordinate systems of the spherical and cuboid magnets, respectively, for intensity calculation. To obtain the magnetic field intensity at an arbitrary given point in the field of spherical and cuboid magnets, we first need to transfer the point from the environment coordinate system to the magnetic field coordinate system of the spherical and cuboid magnets. Deakin [54] has introduced the form of 3-D conformal coordinate transformation that combines axes rotation, scale change, and origin shifts. Analogously, our coordinate system transformation of points in space is a Euclidean transformation that only consists of rotation and translation, which can be defined as

$$\begin{bmatrix} p_x^m \\ p_y^m \\ p_z^m \end{bmatrix} = R_{\theta_x} R_{\theta_y} R_{\theta_z} \begin{bmatrix} p_x^e \\ p_y^e \\ p_z^e \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{6}$$

where p_x^m, p_y^m, p_z^m and p_x^e, p_y^e, p_z^e are 3-D coordinates of the point in the magnetic field coordinate system and environment coordinate system, respectively. T_x, T_y, and T_z are translations between the origins of the two coordinate systems along the x-axis, y-axis, and z-axis. R_{θ_x}, R_{θ_y}, and R_{θ_z} are the rotation matrices generated by the rotation angle of the magnetic field coordinate system with respect to the environment coordinate system around the x-axis, y-axis, and z-axis, which can be expressed by

$$\begin{cases} R_{\theta_x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \\ R_{\theta_y} = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \\ R_{\theta_z} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{cases} \tag{7}$$

where the positive sense of rotation angle is determined by the right-hand screw rule.



(a) Spherical magnet (b) Cuboid magnet

Fig. 2. Magnetic field coordinate systems of the spherical and cuboid magnets, in which $P_s(x_s, y_s, z_s)$ and $P_c(x_c, y_c, z_c)$ are arbitrary given points.

Now we have implemented the 3-D conformal coordinate transformation. Then we will illustrate how to calculate the magnetic field intensity of the points in the magnetic field coordinate system. For the spherical permanent magnet, given an arbitrary point (x_s, y_s, z_s) in its magnetic field coordinate system, we first need to convert the point from Cartesian coordinates to Spherical coordinates by the formulae we defined below:

$$\begin{cases} r_s = \sqrt{x_s^2 + y_s^2 + z_s^2} \\ \theta_s = \arccos \dfrac{z_s}{r_s + \epsilon} \\ \phi_s = \arctan \dfrac{y_s}{x_s + \epsilon} \end{cases} \tag{8}$$

where r_s, θ_s, and ϕ_s denote the radius, inclination, and azimuth in Spherical coordinates, respectively, and ϵ is a small constant. Let a_s denote the radius of the spherical magnet, then the magnetic field intensity of the spherical magnet at given point (r_s, θ_s, ϕ_s) can be calculated analytically as

$$\begin{cases} H_x^s = \dfrac{M_s}{4\pi} \displaystyle\int_0^{\pi} \int_0^{2\pi} [a_s^3 \sin^3\theta_0 \cos\varphi_0 (r_s \cos\theta_s - a_s \cos\theta_0)] \\ \qquad /[(r_s^2 + a_s^2 - 2a_s r_s (\cos\theta_s \cos\theta_0 \\ \qquad + \sin\theta_s \sin\theta_0 \cos(\varphi_s - \varphi_0)))^{3/2} + \epsilon] d\varphi_0 d\theta_0 \\ H_y^s = \dfrac{M_s}{4\pi} \displaystyle\int_0^{\pi} \int_0^{2\pi} [a_s^3 \sin^3\theta_0 \sin\varphi_0 (r_s \cos\theta_s - a_s \cos\theta_0)] \\ \qquad /[(r_s^2 + a_s^2 - 2a_s r_s (\cos\theta_s \cos\theta_0 \\ \qquad + \sin\theta_s \sin\theta_0 \cos(\varphi_s - \varphi_0)))^{3/2} + \epsilon] d\varphi_0 d\theta_0 \\ H_z^s = \dfrac{M_s}{4\pi} \displaystyle\int_0^{\pi} \int_0^{2\pi} [a_s^3 \sin^3\theta_0 (a_s \sin\theta_0 - r_s \cos\theta_s \\ \qquad \cos(\varphi_s - \varphi_0))]/[(r_s^2 + a_s^2 - 2a_s r_s (\cos\theta_s \cos\theta_0 \\ \qquad + \sin\theta_s \sin\theta_0 \cos(\varphi_s - \varphi_0)))^{3/2} + \epsilon] d\varphi_0 d\theta_0 \\ H_s = \sqrt{(H_x^s)^2 + (H_y^s)^2 + (H_z^s)^2} \end{cases} \tag{9}$$

where M_s is the magnetization intensity of the spherical magnet. H_x^s, H_y^s, and H_z^s are the calculated intensities of the magnetic field along the x-axis, y-axis, and z-axis for a spherical magnet, and H_s is the required value of magnetic field intensity for the given point (x_s, y_s, z_s) in the field of the spherical magnet.

For the cuboid magnet, let l_c, w_c, h_c denote the length, width, and height of the cuboid magnet along the x-, y-, and z-axis, respectively. Then the magnetic field intensity at a given point (x_c, y_c, z_c) can be calculated as

$$\begin{cases} H_x^c = -\dfrac{M_c}{8\pi} [\Gamma(l_c - x_c, w_c - y_c, z_c) + \Gamma(l_c - x_c, y_c, z_c) \\ \qquad - \Gamma(x_c, w_c - y_c, z_c) - \Gamma(x_c, y_c, z_c)] \\ H_y^c = -\dfrac{M_c}{8\pi} [\Gamma(w_c - y, l_c - x_c, z_c) + \Gamma(w_c - y_c, x_c, z_c) \\ \qquad - \Gamma(y_c, l_c - x_c, z_c) - \Gamma(y_c, x_c, z_c)] \\ H_z^c = -\dfrac{M_c}{8\pi} [\Psi(w_c - y_c, l_c - x_c, z_c) + \Psi(y_c, l_c - x_c, z_c) \\ \qquad + \Psi(l_c - x_c, w_c - y_c, z_c) + \Psi(x_c, w_c - y_c, z_c) \\ \qquad + \Psi(w_c - y_c, x_c, z_c) + \Psi(y_c, x_c, z_c) \\ \qquad + \Psi(l_c - x_c, y_c, z_c) + \Psi(x_c, y_c, z_c)] \\ H_c = \sqrt{(H_x^c)^2 + (H_y^c)^2 + (H_z^c)^2} \end{cases} \tag{10}$$

where M_c denotes the magnetization intensities of the cuboid magnet. H_x^c, H_y^c, and H_z^c are the calculated intensities of the magnetic field along the x-axis, y-axis, and z-axis for a cuboid magnet, and H_c is the required value of the magnetic field intensity for the given point (x_c, y_c, z_c) in the field of the cuboid magnet. $\Gamma(\gamma_1,\gamma_2,\gamma_3)$ and $\Psi(\varphi_1,\varphi_2,\varphi_3)$ are two auxiliary functions to simplify calculations as defined by

$$\begin{cases}\Gamma(\gamma_1,\gamma_2,\gamma_3)=\ln\left(\dfrac{\sqrt{(\gamma_1)^2+(\gamma_2)^2+(\gamma_3-z_0)^2}-\gamma_2}{\sqrt{(\gamma_1)^2+(\gamma_2)^2+(\gamma_3-z_0)^2}+\gamma_2+\epsilon}+\epsilon\right)\Bigg|_{z_0=0}^{z_0=h_c}\\ \Psi(\varphi_1,\varphi_2,\varphi_3)=\arctan\dfrac{\varphi_1(\varphi_3-z_0)}{\varphi_2\sqrt{(\varphi_1)^2+(\varphi_2)^2+(\varphi_3-z_0)^2}+\epsilon}\Bigg|_{z_0=0}^{z_0=h_c}.\end{cases} \tag{11}$$

With these analytical expressions, we are now able to obtain the specific value of magnetic field intensity at any given point in the field of spherical magnet and cuboid magnet. For simplicity, we let $M_s = M_c = 4\pi$ as the value of magnetization intensity only determines the magnitude of magnetic field intensity, rather than its distribution. Fig. 3 visualizes the distributions of magnetic field intensity generated by the spherical and cuboid magnets in 3-D space, where the sphere's radius is 0.02, the length, width, and height of the cuboid are 0.1, 0.4, 0.05, respectively. Areas with warmer colors have a higher intensity value in the magnetic field. It can be observed that the surfaces of the sphere and cuboid magnets exhibit the highest intensity value, and the intensity decreases intensively as the distance to the magnet increases.

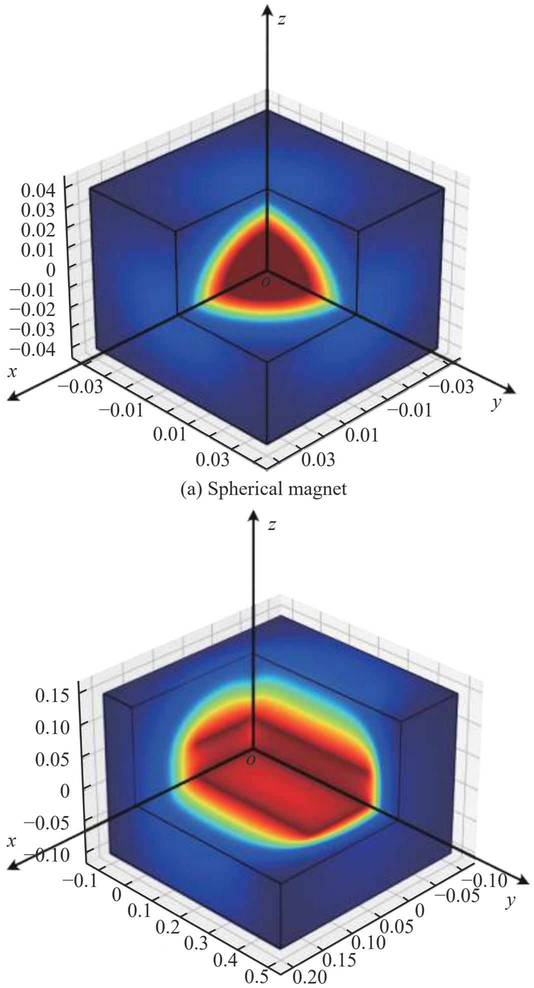
Towards the environments with higher dimensions (e.g., more than three dimensions), though we cannot directly calculate the magnetic field intensity since the target/obstacles are not physical entities and thus can not be considered as permanent magnets in the physical sense, the concept and properties of the magnetic field can still be extended to the scenario with a high-dimensional goal in a mathematical sense. For instance, the magnetic field intensity of a spherical permanent magnet along the axis of magnetization (taking the z-axis as an example) can be calculated as below:

$$H=\begin{cases}\dfrac{2M_0r^3}{3|z|^3}\hat{z}, & (|z|\ge r)\\ -\dfrac{M_0}{3}\hat{z}, & (|z|<r)\end{cases} \tag{12}$$

where M_0 is the magnetization intensity, r is the radius of the spherical magnet, and z denotes the coordinate on the z-axis in the magnetic field coordinate system. As we can see, the intensity value is inversely proportional to the third power of the distance to the origin of magnetic field coordinate system along the magnetization axis outside the magnet. This property of the spherical magnets can be extended to high-dimensional goals by simply replacing $|z|$ with a distance metric (e.g., L2-norm) between the current state and the target state, which can still maintain a nonlinear and intensively-varying distribution of shaping rewards that provides sufficient information about the optimization landscape.

C. Magnetic Reward Function

Considering a goal-conditioned task with 1 target and N obstacles. Let $\mathcal{M}_T(P_a^T)$ and $\mathcal{M}_{O_i}(P_a^{O_i})$ $(i=1,2,\ldots,N)$ denote the functions that calculate the intensity values H_T and H_{O_i} of the magnetic fields generated by the target magnet and obstacle magnets, respectively, where P_a^T and $P_a^{O_i}$ are the agent's positions in the magnetic field coordinate system of target and obstacle magnets. Taking the spherical target and cuboid obstacle as an example, the intensity functions $\mathcal{M}_T$ and $\mathcal{M}_O$ can be defined as (9) and (10) in Section III-B, respectively. The intensity values cannot be directly combined since the scales of magnetic field intensity for various magnets are different using separate ways of calculation. It may amplify the effect of one or some magnets and give an incorrect reward signal to the agent if the intensity scales of these magnets are much larger than others. Therefore, we need to standardize the calculated intensity values H_T and H_{O_i}.



(a) Spherical magnet

(b) Cuboid magnet

Fig. 3. Distributions of the magnetic field intensity in 3-D space generated by the spherical and cuboid magnets. Areas painted with warmer color possess a larger intensity value of the magnetic field.

Intuitively, we use the z-score method to unify the unevenly distributed intensity values into a standard Gaussian distribution for the target and obstacles magnets as

$$\begin{cases}H_T^N=\dfrac{H_T-\mu_T}{\sigma_T+\epsilon}\\ H_{O_i}^N=\dfrac{H_{O_i}-\mu_{O_i}}{\sigma_{O_i}+\epsilon} \quad (i=1,2,\ldots,N)\end{cases} \tag{13}$$

where ϵ is a small constant, μ_T and μ_{O_i} denote the means of magnetic field intensity for the target and obstacles magnet, and σ_T and σ_{O_i} denote the standard deviation. Unfortunately, the actual values of the mean and standard deviation cannot be obtained directly since the agent knows nothing about the specific intensity distribution at the beginning of training. Hence, we turn to estimate these values by introducing an extra magnet buffer $\mathcal{D}_\mathcal{M}$ that stores the calculated intensity values H_T and H_{O_i}. During the learning process, we concurrently update the values of the mean and standard deviation for H_T and H_{O_i} using the up-to-date buffer $\mathcal{D}_\mathcal{M}$, which are used to standardize the values of calculated magnetic field intensity in the next learning episode. According to the law of large numbers, as the number of collected data in $\mathcal{D}_\mathcal{M}$ increases, the estimated values of the mean and standard deviation will converge to their actual values.

After implementing the standardization of the magnet field intensity, the unified intensity of the target and obstacles magnets can be combined to a value that represents the overall intensity of the magnetic field generated by the target and obstacles together, which can be expressed as

$$H_{com} = H_T^N - \frac{1}{N}\sum_{i=1}^{N} H_{O_i}^N \quad (i = 1, 2, \ldots, N). \tag{14}$$

Recall our motivation in Section III-A, we want the agent to be attracted by the target and repelled by the obstacles when exploring the environment. In addition, we consider that the target's attraction has the exact extent of the effect as all obstacles' repulsion. Accordingly, we take the mean of unified intensity values among all the obstacles in the calculation and give a positive value to H_T^N while a negative value to the mean of $H_{O_i}^N$ to define the combined intensity value H_{com}.

Due to the physical property of magnets, the intensity values will be tremendous in the near region of the target, as shown in Fig. 1. Therefore, regarding H_{com} as the magnetic reward could probably lead to a phenomenon that the value of the magnetic reward is orders of magnitude larger than the original reward. To address the value exploding problem, we employ the Softsign function to normalize the combined intensity value within a bounded range of $[-1, 1]$ and use the output of the Softsign function as the magnetic reward R^m

$$R^m = \text{Softsign}(H_{com}) = \frac{H_{com}}{1 + |H_{com}|}. \tag{15}$$

Algorithm 1 Magnetic Reward Function

```
Input: state $s$; goal $g$;
    intensity calculation functions $\mathcal{M}_T, \mathcal{M}_{O_i}$;
    $\mu_T, \sigma_T, \mu_{O_i}, \sigma_{O_i}$ $(i = 1, 2, \ldots, N)$
Output: magnetic field intensity $H_T, H_{O_i}$;
    magnetic reward $R^m$
Obtain agent's position $P_a$ according to $s$
Obtain target's position $P_T$ and obstacles' positions $P_{O_i}$ according to $g$
Transfer $P_a$ to $P_a^T, P_a^{O_i}$ using (6) according to $P_T, P_{O_i}$
$H_T \leftarrow \mathcal{M}_T(P_a^T), H_{O_i} \leftarrow \mathcal{M}_{O_i}(P_a^{O_i})$
Calculate $H_T^N$ and $H_{O_i}^N$ using (13) according to $\mu_T, \sigma_T, \mu_{O_i}, \sigma_{O_i}$
Calculate $H_{com}$ using (14)
$R^m \leftarrow \text{Softsign}(H_{com})$
```

Algorithm 1 shows the calculation process of the magnetic reward function in our method. Since we consider the obstacles as a part of the goal in goal-conditioned RL together with the target, we define the goal in our setting as $g := [P_T, P_{O_1}, P_{O_2}, \ldots, P_{O_N}]$, where P_T is the target's position and $P_{O_1}, P_{O_2}, \ldots, P_{O_N}$ are obstacles' positions.

D. Magnetic Field-Based Reward Shaping

After calculating the magnetic reward R^m that represents the overall intensity generated by the target and obstacles magnets with normalization techniques, we need to guarantee the optimal policy invariance of our method when using R^m as the shaping reward. PBRS [26] defines the shaping reward function as $F(s, a, s) = \gamma\Phi(s) - \Phi(s)$ with proof of sufficiency and necessity that the optimal policy remains unchanged, where $\Phi(\cdot)$ denotes the potential function carrying human domain knowledge, and $\gamma \in [0, 1]$ is the discount factor. In this paper, we employ the DPBA [29] method to transform our magnetic reward R^m into the form of potential-based reward shaping in the goal-conditioned RL setting. To be specific, we need to achieve $F \approx R^m$, where the potential function in F can be learned in an on-policy way [55] with a technique analogous to SARSA [56]

$$\Phi_{t+1}(s, a, g) \leftarrow \Phi_t(s, a, g) + \eta\delta_t^\Phi \tag{16}$$

where η is the learning rate of this secondary goal-conditioned potential function Φ, and δ_t^Φ denotes the temporal difference (TD) error of the state transition

$$\delta_t^\Phi = r_t^\Phi + \gamma\Phi_t(s_{t+1}, a_{t+1}, g) - \Phi_t(s_t, a_t, g) \tag{17}$$

where a_{t+1} is chosen using the current policy. According to DPBA [29], the value of R^Φ equals the negation of the expert-provided reward function, which is the magnetic reward in our method: $R^\Phi = -R^m$, and thus $r_t^\Phi = -r_t^m$.

Algorithm 2 MFRS for Goal-Conditioned RL

```
Input: potential function $\Phi_\psi$; replay buffer $\mathcal{D}_\mathcal{R}$; magnet buffer $\mathcal{D}_\mathcal{M}$; number of obstacles $N$; discount factor $\gamma$; off-policy RL algorithm $\mathbb{A}$
Output: optimal goal-conditioned policy $\pi_\theta^*$
Initialize $\pi_\theta$ arbitrarily, $\Phi_\psi \leftarrow 0$
Initialize replay buffer $\mathcal{D}_\mathcal{R}$ and magnet buffer $\mathcal{D}_\mathcal{M}$
$\mu_T, \mu_{O_i} \leftarrow 0$; $\sigma_T, \sigma_{O_i} \leftarrow 1$ $(i = 1, 2, \ldots, N)$
for episode = 1, E do
    Sample an initial state $s_0$ and a goal $g$
    for t = 0, H - 1 do
        $a_t \leftarrow \pi_\theta(s_t, g)$
        Execute $a_t$, observe $s_{t+1}$ and $r_t$
        Calculate $H_T, H_{O_i}$ and $r_t^m$ using Algorithm 1 according to $s_{t+1}, g$ and $\mu_T, \sigma_T, \mu_{O_i}, \sigma_{O_i}$
        Store $H_T, H_{O_i}$ in $\mathcal{D}_\mathcal{M}$
        $a_{t+1} \leftarrow \pi_\theta(s_{t+1}, g)$
        Update $\psi$ using (19)
        Calculate $f_t$ using (20)
        $r_t' \leftarrow r_t + f_t$
        Store transition $(s_t, a_t, s_{t+1}, r_t', g)$ in $\mathcal{D}_\mathcal{R}$
    end
    Update $\pi_\theta$ using $\mathbb{A}$
    $\mu_T \leftarrow mean(H_T), \sigma_T \leftarrow std(H_T)$
    $\mu_{O_i} \leftarrow mean(H_{O_i}), \sigma_{O_i} \leftarrow std(H_{O_i})$
end
```

Since we focus on the tasks with continuous state and action space, the goal-conditioned potential function can be parameterized using a deep neural network $\Phi_\psi(s,a,g)$ with the weights $\boldsymbol{\psi}$. Akin to the temporal-difference learning, we use the Bellman error as the loss function of this potential network as

$$\mathcal{L}_\Phi = \frac{1}{2}\Big((-r_t^m + \gamma\Phi_\psi(s_{t+1},a_{t+1},g) - \Phi_\psi(s_t,a_t,g)\Big)^2. \quad (18)$$

Hereafter, the weights of $\Phi_\psi(s,a,g)$ can be updated using the gradient descent as

$$\psi \leftarrow \psi - \eta\nabla_\psi\mathcal{L}_\Phi. \quad (19)$$

As the secondary goal-conditioned potential function Φ is updated every timestep, the potential-based shaping reward F at each timestep can be expressed as

$$f_t = \gamma\Phi_{t+1}(s_{t+1},a_{t+1},g) - \Phi_t(s_t,a_t,g). \quad (20)$$

Algorithm 2 presents the integrated process of MFRS for goal-conditioned RL. Next, we give a theoretical guarantee of optimal policy invariance in the goal-conditioned RL setting of our method in the Appendix, and a convergence analysis that the expectation of shaping reward F will be equal to our magnetic reward R^m when the goal-conditioned potential function Φ has converged in Theorem 1 below.

Theorem 1: Let Φ be the goal-conditioned potential function updated by (16) with the state transition matrix $\mathcal{T}$, where R^Φ equals the negation of our magnetic reward R^m. Then, the expectation of shaping reward F expressed in (20) will be equal to R^m when Φ has converged.

Proof: The goal-conditioned potential function Φ follows the update rule of the Bellman Equation [2], which enjoys the same recursive relation when the potential value has converged to the TD-fixpoint Φ^*. Thus, we have

$$\begin{aligned}\Phi^*(s,a,g) &= R^\Phi(s,a,g) + \gamma\mathbb{E}[\Phi^*(s,a',g)]\\ &= -R^m(s,a,g) + \gamma\mathbb{E}[\Phi^*(s,a',g)]. \quad (21)\end{aligned}$$

According to (20), we have the shaping reward F with respect to the converged potential value Φ^*

$$\begin{aligned}F(s,a,s,a',g) &= \gamma\Phi^*(s,a',g) - \Phi^*(s,a,g)\\ &= \gamma\Phi^*(s,a',g) + R^m(s,a,g)\\ &\quad - \gamma\mathbb{E}[\Phi^*(s,a',g)]\\ &= R^m(s,a,g) + \gamma(\Phi^*(s,a',g)\\ &\quad - \mathbb{E}[\Phi^*(s,a',g)]). \quad (22)\end{aligned}$$

To obtain the expected shaping reward $\mathcal{F}(s,a,g)$, we take the expectation with respect to the state transition matrix $\mathcal{T}$

$$\begin{aligned}\mathcal{F}(s,a,g) &= \mathbb{E}[F(s,a,s,a',g)]\\ &= R^m(s,a,g) + \gamma\mathbb{E}\left[\Phi^*(s,a',g) - \mathbb{E}[\Phi^*(s,a',g)]\right]\\ &= R^m(s,a,g). \quad (23)\end{aligned}$$

Hence, the expectation of shaping reward F will be equal to our magnetic reward R^m when the goal-conditioned potential function Φ converges to the TD-fixpoint. ■

IV. Simulation Experiments

To evaluate our method, we build a 3-D simulated robotic manipulation platform based on the software CoppeliaSim, formerly known as V-REP [57], and design a set of challenging goal-conditioned RL tasks with continuous state-action space. Being analogous to the FetchReach-v1 [58] task in OpenAI Gym [59], our tasks require the agent to move the end-effector of a robotic arm to dynamic targets with different positions, but differ from FetchReach-v1 in additional dynamic obstacles that the agent should avoid.

A. Environment Settings

Sections IV-B and IV-C present the results and insightful analysis of our findings. In the experiments, we evaluate MFRS in comparison to several baseline methods as follows:

1) No Shaping (NS): As the basic sparse reward condition, it trains the policy with the original reward given by the environment without shaping.

2) PBRS [26]: It adds a calculated shaping reward $F = \gamma\Phi(s,g) - \Phi(s,g)$ to the original reward, where the potential function is built on Euclidean distance as $\Phi(s,g) = -d(p_a,p_g) + \frac{1}{N}\sum_{i=1}^N d(p_a,p_{o_i})$.

3) DPBA [29]: It adds a learned shaping reward $F = \gamma\Phi_{t'}(s,a',g) - \Phi_t(s,a,g) \approx R^\dagger$ to the original reward, where the objective reward function is built on Euclidean distance as $R^\dagger = -d(p_a,p_g) + \frac{1}{N}\sum_{i=1}^N d(p_a,p_{o_i})$.

4) HER [23]: As the famous relabeling strategy in goal-conditioned RL, it relabels the desired goals in the replay buffer with the achieved goals in the same trajectories. In our experiments, we adopt the default "final" strategy in HER that chooses the additional goals corresponding to the final state of the environment.

5) Sibling Rivalry (SR) [30]: It samples two sibling episodes for each goal simultaneously and uses each others' achieved goals as anti-goals $\overline{g}$. Then, it engineers an additional reward bonus $d(s,\overline{g})$ to encourage exploring and avoid local optima, where d is the Euclidean distance.

6) Adversarial Intrinsic Motivation (AIM) [42]: It aims to learn a goal-conditioned policy whose state visitation distribution minimizes the Wasserstein distance to a target distribution for a given goal, and utilizes the Wasserstein distance to formulate a shaped reward function, resulting in a nonlinear reward shaping mechanism.

We use Dobot Magician [60] as the learning agent in our experiments, which is a 3-degree of freedom (DOF) robotic arm in both CoppeliaSim simulated environment and real-world industrial applications. Dobot Magician has three stepper motors to actuate its joints J_1, J_2, and J_3, as shown in Fig. 4, which can achieve rotation angles within the range of $[-90°,90°],[0°,85°],[-10°,90°]$, respectively. Therefore, we consider the action as a 3-dimensional vector clipped to be in the range of $[-1°,1°]$, representing the rotation velocity of these three joints. Being analogous to FetchReach-v1, the observations include the Cartesian positions of the "elbow", "finger", and the rotation angles of the three joints.

In our experiment, we use deep deterministic policy gradient (DDPG) [61] as the base algorithm to evaluate all the investigated methods under the same configurations in the goal-conditioned RL setting. Specifically, the goal-condi-

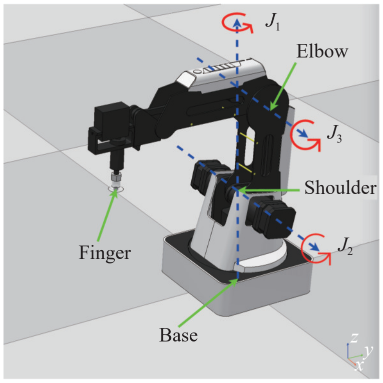


Fig. 4. The configuration of Dobot Magician, where "base", "shoulder", and "elbow" represent three motors with the joints denoted by J_1, J_2, and J_3, respectively, and "finger" represents the end-effector.

tioned policy is approximated by a neural network with two 256-unit hidden layers separated by ReLU nonlinearity as the actor, which maps each state and goal to a deterministic action and updates its weights using Adam optimizer [62] with a learning rate of $\alpha = 3 \times 10^{-4}$. The goal-conditioned Q-function is also approximated by a neural network that maps each state-action pair and goal to a certain value as the critic, which has the same network structure and optimizer as the actor with a learning rate of $\beta = 10^{-3}$. The target networks of actor and critic update their weights by slowly tracking the learned actor and critic with $\tau = 0.001$. We train the DDPG whenever an episode terminates with a minibatch size of 128 for 100 iterations. In addition, we use a replay buffer size of 10^6, a discount factor of $\gamma = 0.99$, and a Gaussian noise $\mathcal{N}(0, 0.4)$ to each action for exploration. For MFRS and DPBA methods, the goal-conditioned potential function Φ is approximated by a neural network with the same structure and optimizer as the critic network and a learning rate of $\eta = 10^{-4}$, which is updated for each timestep in an on-policy manner. For MFRS in particular, we set the magnet buffer size of 10^6 and the small constant of $\epsilon = 10^{-7}$.

For each report unit, we define two performance metrics. One is the success rate in each learning episode, where success is defined as the agent reaching the target without hitting on the obstacles. The other is the average timesteps for each trajectory over all learning episodes, defined as $\frac{1}{E}\sum_{k=1}^{E} T_k$, where E is the number of learning episodes and T_k is the terminal timestep in the kth episode. The former is plotted in figures, and the latter is presented in tables. Moreover, due to the randomness in the update process of the neural network, we repeat five runs for each policy training by taking different random seeds for all methods, and report the mean regarding the performance metrics. Our code is available online[1].

B. Primary Results

To evaluate our method in different scenarios, we develop four goal-conditioned tasks with various dynamics configurations of the target and obstacles, on which we implement MFRS and baseline methods. All four tasks require the agent to move the "finger" into a target region without hitting the obstacles in each episode. Accordingly, we define the original reward function as: $r(s, a, g) = 100$ if the "finger" successfully reaches the target, $r(s, a, g) = -10$ if the agent hits the obstacle or the floor in CopperliaSim, and $r(s, a, g) = -1$ otherwise. Each episode starts by sampling the positions of the target and obstacles with the rotation angles of the agent's three joints reset to zero, and terminates whenever the agent reaches the target or at the horizon of $H = 1000$.

1) Goal-Conditioned Tasks With Different Goal Dynamics: In the first two tasks, the obstacle is a cuboid rotator revolving around a fixed cylindrical pedestal, where the length, width, and height of the rotator are 0.1, 0.4, 0.05 along the x-axis, y-axis, z-axis, and the radius, height of the pedestal are 0.04, 0.13, respectively. The rotator's axis coincides with the pedestal's axis, and the top surfaces of the rotator and pedestal are on the same plane. We represent the obstacle position by the center point coordinate of the rotator determined by its rotation angle from an initial position, where the positive direction of rotation is defined to be anticlockwise, with the rotator's initial position being parallel to the y-axis. In the other two tasks with higher complexity, the agent has to handle three dynamic spherical obstacles with a radius of 0.02, 0.04, and 0.06, respectively. We represent the obstacles' positions by the center point coordinates of their own spheres. Besides, the target is a sphere with a radius of 0.02 in all four tasks where the target position is the center point coordinate.

i) Task I: As shown in Fig. 5(a), the dynamic of the goal is determined by randomly changing the target position, which is sampled uniformly from the space both in the reachable area of "finger" and below but not under the lower surface of the single static obstacle.

ii) Task II: As shown in Fig. 5(b), the dynamic goal is created by changing the positions for both the target and the single obstacle. The change of obstacle position can be converted into sampling rotation angles of the rotator, which is defined as a uniform sampling within $[-60°, 60°]$. The change of target position is consistent with the one in Task I.

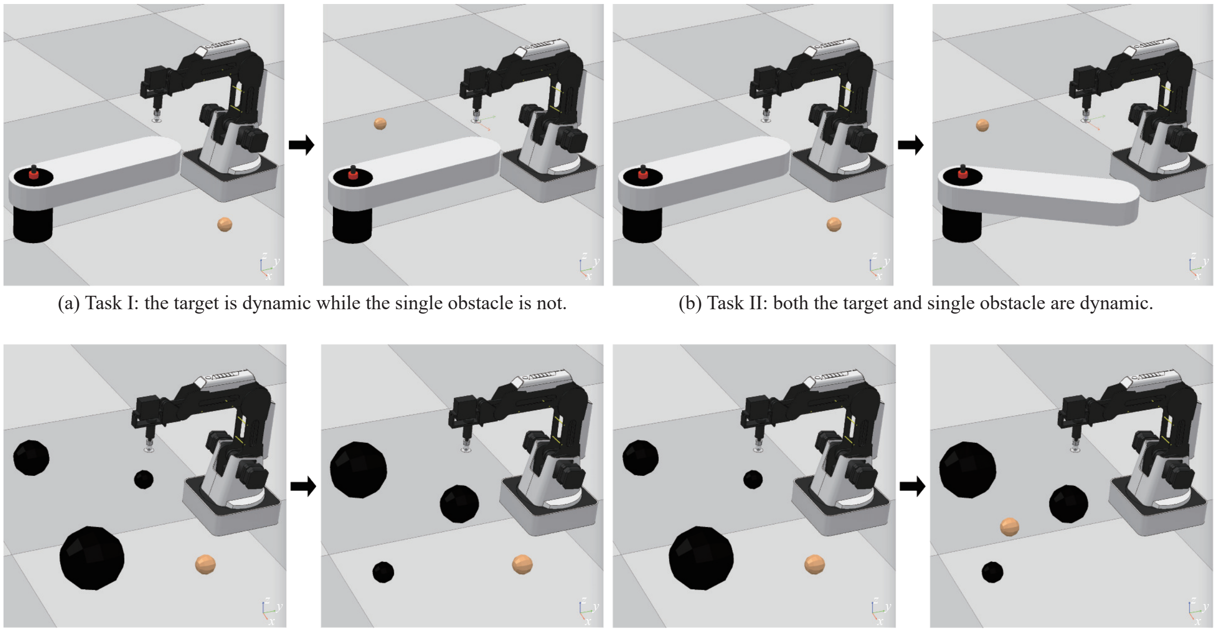
iii) Task III: As shown in Fig. 5(c), the goal dynamics is determined by changing the positions of multiple obstacles while holding the target still, and all the target and obstacles are ensured to be not intersectant.

iv) Task IV: As shown in Fig. 5(d), this kind of dynamic goal is created by sampling the positions of all the target and multiple obstacles simultaneously, which is considered the most complex of the four tasks. Also, we need to ensure that all the target and obstacles are not intersectant.

2) Results of MFRS: We present the experimental results of MFRS and all baselines implemented on the four goal-conditioned tasks. Fig. 6 shows the success rate in each learning episode, of which the mean across five runs is plotted as the bold line with 90% bootstrapped confidence intervals of the mean painted in the shade. Table I reports the numerical results in terms of average timesteps over 10 000 learning episodes of all tested methods. The mean across five runs is presented, and the confidence intervals are corresponding standard errors. The best performance is marked in boldface.

Surprisingly, HER obtains the worst performance and even

[1] https://github.com/Darkness-hy/mfrs



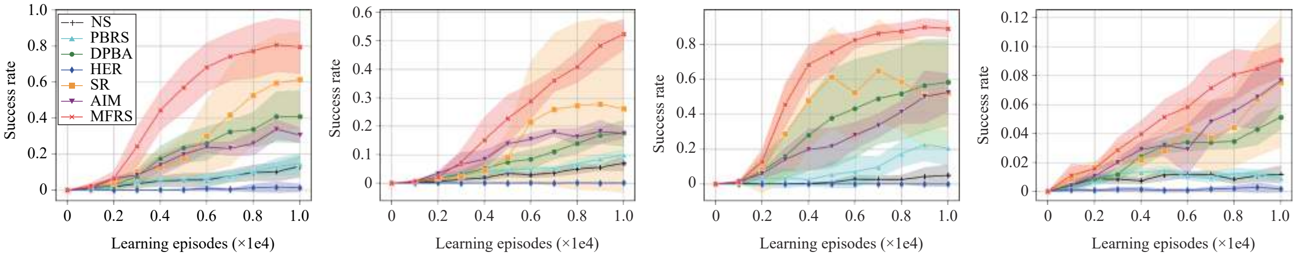
(a) Task I: the target is dynamic while the single obstacle is not.

(b) Task II: both the target and single obstacle are dynamic.

(c) Task III: the multiple obstacles are dynamic while the target is not.

(d) Task IV: both the target and multiple obstacles are dynamic.

Fig. 5. Examples of four goal-conditioned tasks with different dynamics of the target and obstacles. The spherical targets in all the tasks are shown in yellow, the single cuboid obstacles in Tasks I and II are shown in white, and the multiple spherical obstacles in Tasks III and IV with various sizes are shown in black.



(a) Task I

(b) Task II

(c) Task III

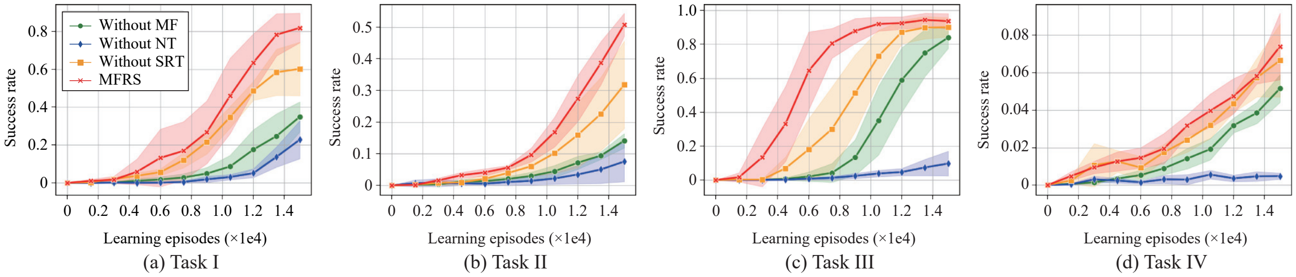
(d) Task IV

Fig. 6. Success rate in each learning episode of all investigated methods implemented on four different goal-conditioned tasks.

TABLE I
NUMERICAL RESULTS IN TERMS OF AVERAGE TIMESTEPS OVER ALL LEARNING EPISODES OF TESTED METHODS IN FOUR DIFFERENT GOAL-CONDITIONED TASKS

Task	Task I	Task II	Task III	Task IV
NS	939.9 ± 12.7	970.0 ± 4.0	982.8 ± 9.0	990.9 ± 0.9
PBRS	933.1 ± 9.9	952.1 ± 2.7	902.8 ± 29.4	989.9 ± 0.4
DPBA	785.3 ± 31.0	916.5 ± 83.6	660.7 ± 86.3	972.7 ± 1.7
HER	993.4 ± 4.8	999.0 ± 106.2	1000.0± 0.0	998.6 ± 0.5
SR	684.3 ± 78.3	842.0 ± 78.9	524.5± 111.2	962.9 ± 9.5
AIM	807.8 ± 7.5	859.9 ± 9.4	731.6± 17.7	960.1 ± 1.9
MFRS	**513.1 ± 46.4**	**760.8 ± 62.0**	**380.5 ± 18.7**	**948.6 ± 4.3**

gets a lower success rate than NS. We conjecture that the relabeling strategy may mislabel the obstacles as the targets when considering the obstacles as a part of the goal. PBRS achieves a slightly higher success rate than NS since it incorporates additional distance information into the potential function. DPBA obtains better performance than PBRS, indicating that directly encoding the distance information as the shaping reward and learning a potential function in turn can be more informative and efficient. AIM has similar performance compared to DPBA and performs better in complex tasks (Tasks II and IV), which is supposed to benefit from the estimated Wasserstein-1 distance between the state visitation distribution and target distribution. SR performs best among the baseline methods on account of its anti-goals mechanism that encourages exploring and avoids local optima, while it suffers from large confidence intervals since SR fails to consider obstacles and can not guarantee the optimal policy invariance property.



(a) Task I (b) Task II (c) Task III (d) Task IV

Fig. 7. Success rate in each learning episode of the three variants together with MFRS implemented on four different goal-conditioned tasks.

In contrast, it can be observed from Fig. 6 that MFRS outperforms all the baseline methods with a considerably larger success rate in all the four tasks, which is supposed to benefit from the explicit and informative optimization landscape provided by the magnetic field intensity. The performance gap in terms of success rate at the end of training varies in different tasks. For instance, in Task I, MFRS achieves a success rate of 79.66% at the last learning episode while the baseline methods have only increased the success rate to no more than 61.32%, which is 29.91% improved performance at least. Likewise, the performance in terms of success rate is improved by 100.38% in Task II, 52.85% in Task III and 18.59% in Task IV. In addition, being different from the research efforts on goal-conditioned RL, our method aims at using reward shaping method to solve sparse reward problems for goal-conditioned RL, which is capable of extending the goal setting to not only dynamic target but also dynamic obstacles while holding the optimal policy invariance property at the same time. Moreover, from the large performance gap in terms of success rate between MFRS and AIM, we can deduce that MFRS has the potential to maintain the superiority over other nonlinear reward shaping mechanisms.

From Table I, it can be obtained that MFRS achieves significantly smaller average timesteps over 10 000 learning episodes than all the baselines in all the tasks, which means fewer sampled transitions will be required for the agent to learn the policy. Specifically, in Task I, MFRS achieves 513.1 timesteps in average over all learning episodes while the NS method acquires 939.9 timesteps. It indicates that MFRS obtains a 45.4% reduction of sampling transitions. The decrease of timesteps is 21.6% in Task II, 61.3% in Task III, and 4.3% in Task IV. Hence, our method successfully improves the sample efficiency of RL in the goal-conditioned setting. In summary, being consistent with the statement in Section III-A, it is verified that MFRS is able to provide sufficient and conducive information about the complex environments with various dynamics of the target and obstacles, achieving significant performance for addressing the sparse reward problem in different scenarios.

C. Ablation Study

To figure out how the three critical components of our method affect the performance respectively, we perform additional ablation studies using a control variables approach to separate each process apart as the following variants of MFRS.

1) Without MF (Magnetic Field): The shaping reward is calculated based on Euclidean distance, instead of the magnetic field, with normalization techniques and converted into the form of potential-based reward shaping.

2) Without NT (Normalization Techniques): The shaping reward is calculated based on the magnetic field without normalization techniques and directly converted into the potential-based form.

3) Without SRT (Shaping Reward Transformation): The shaping reward directly follows the form of potential-based reward shaping, with the potential value calculated according to the magnetic field with normalization techniques.

The learning performance in terms of success rate per episode of the three variants as well as MFRS is shown in Fig. 7.

First and foremost, the variant *without MF* is compared to MFRS to identify how magnetic field-based shaping reward improves the sample efficiency against distance-based setting. We can observe that MFRS outperforms the variant *without MF* with a considerably larger success rate per episode in all four tasks. Specifically, in Task III, it takes the variant *without MF* 15 000 episodes to achieve a success rate of around 80%, while MFRS only needs 8000 episodes to do that. It verifies that magnetic field-based shaping reward is able to provide a more explicit and informative optimization landscape for policy learning than the distance-based setting. It is consistent with the statement in Section I that the nonlinear and anisotropic properties of the generated magnetic field provide a sophisticated reward function that carries more accessible and sufficient information about the optimization landscape, thus resulting in a sample-efficient method for goal-conditioned RL.

Next, the variant *without NT* is compared to MFRS to verify the effectiveness of normalization techniques. Obviously, when taking no account of any normalization in our method, the algorithm performs terribly in all four tasks, which is worse as the task gets more complex. It verifies the assumption in Section III-C that if the intensity scales of some magnets are much larger than others, it may amplify the effect of these magnets and give an incorrect reward signal to the agent. On the other hand, the tremendous intensity value in the very near region of the target will also exacerbate this problem by inducing the agent to slow down the steps to the target, so that it can obtain higher cumulative rewards than directly arriving at the target and terminating the episode.

Finally, the variant *without SRT* is compared to MFRS for verifying the contribution of Shaping Reward Transformation to the performance of our method. From Fig. 7, it can be observed that MFRS outperforms the variant *without SRT* with improved performance in terms of success rate per episode to some extent, and the performance gap is increasing along the learning process, especially in complex tasks. It verifies that the magnetic reward is more informative and effective when approximated by the potential-based shaping reward instead of being regarded as the potential function for the guarantee of optimal policy invariance. On the other hand, one may alternatively select this simplified form of the *without SRT* variant to achieve similar effects of MFRS, which avoids learning a secondary potential function at the cost of some performance loss in practice.

V. Application to Real Robot

To further demonstrate the effectiveness of our method in the real-world environment, we evaluate the performance of MFRS and baselines on a physical Dobot Magician robotic arm. The robot integrates a low-level controller to move the stepper motor of each joint and provides a dynamic link library (DLL) for the user to measure and manipulate the robot's state, including the joint angles and the position of the end-effector. Due to the cost of training on a physical robot, we first train the policies in simulation and deploy them on a real robot without any finetuning. The video is available online[2].

A. Training in Simulation

From the real-world scenario, we consider a goal-conditioned task with a dynamic target and a dynamic obstacle, as shown in Fig. 8(b), and build the same simulated environment accordingly using CoppeliaSim as shown in Fig. 8(a). Both the target and obstacle are in the shape of cuboid. The target cuboid has the length, width, and height of 0.03, 0.045, 0.02 along the x-axis, y-axis, z-axis in simulation, and 0.038, 0.047, 0.12 for obstacle cuboid, respectively. The dynamic of the goal is generated by randomly changing the positions of both the target and obstacle in each episode, which are ensured to be not intersectant. The observations, actions, and reward function are consistent with the ones in Section IV, and all the hyper-parameters are set to be the same as those in Section IV-C.

We present the success rate of MFRS and all the baseline methods training on the real-world task in Fig. 9, of which the mean across 5 runs is plotted as the bold line with 90% bootstrapped confidence intervals of the mean painted in the shade. Being consistent with the experimental results in Section IV-B, MFRS outperforms all other baselines with a significantly larger success rate on the real-world task in simulation.

B. Validation in the Real-World

To figure out whether MFRS performs well and beats the baseline methods on the real robot, we evaluate the success rate and average timesteps over 20 different testing episodes for each run of each method in the training phase, in which the positions of the dynamic target and obstacle are randomly generated and set accordingly. That is, for each involved algorithm in real-world experiments, we have conducted 100 random testing episodes with different settings of the target and obstacle in total.

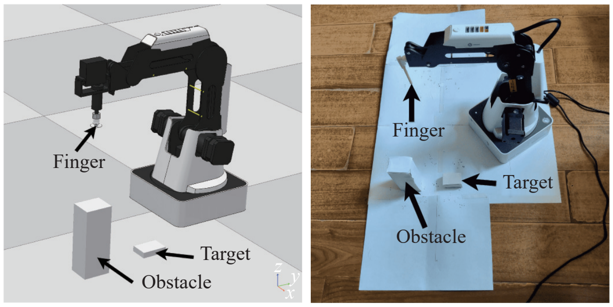


Fig. 8. Illustration of the real-world task in (a) simulation environment; and (b) real-world environment.

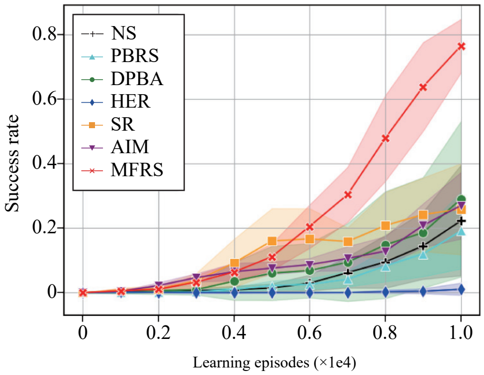


Fig. 9. Success rate in each learning episode of all investigated methods training on the real-world task in simulation environment.

In practice, we read the stepper motors' rotation angles and the end-effector's coordinate by calling the DLL provided by the Dobot Magician, which are then concatenated to form the agent's state. The agent's action in each timestep is obtained by the trained policy, and employed on the real robot as incremental angles to move the three joints using the point-to-point (PTP) command mode of Dobot Magician. During validation, each testing episode starts with the joint angles reset to zero and manually setting the positions of the target and obstacle in the real-world environment that are sampled randomly in simulation, and terminates whenever the "finger" reaches the target, or the robot hits the obstacle or at the horizon of $H = 200$. To prevent unsafe behaviors on the real robot, we restrict the target's sample space to half of the reachable area of the "finger". Computation of the action is done on an external computer, and commands are streamed over the radio at 10Hz using a USB virtual serial port as communication.

We report the numerical results in terms of success rate and average timesteps over 20 testing episodes for each investigated method in Table II, where the mean across five runs using the corresponding policy trained in simulation is presented, and the confidence intervals are the standard errors. From Table II, it can be observed that MFRS achieves a significantly larger success rate and smaller average timesteps

[2] https://hongyuding.wixsite.com/mfrs (or https://www.bilibili.com/video/BV1784y1z7Bj)

TABLE II
NUMERICAL RESULTS IN TERMS OF SUCCESS RATE AND AVERAGE TIMESTEPS OVER 20 TESTING EPISODES OF ALL INVESTIGATED METHODS IN THE REAL-WORLD ENVIRONMENT

Method	Success rate	Average timesteps
NS	0.25 ± 0.21	105.27 ± 48.12
PBRS	0.21 ± 0.15	106.81 ± 47.47
DPBA	0.30 ± 0.29	98.42 ± 51.18
HER	0.01 ± 0.02	147.62 ± 64.17
SR	0.10 ± 0.08	96.74 ± 39.64
AIM	0.28 ± 0.07	86.78 ± 6.28
MFRS	$\mathbf{0.84 \pm 0.07}$	$\mathbf{75.92 \pm 3.46}$

compared to all the baselines on the real robots, which is consistent with the results in the simulation experiment. In addition, our method obtains relatively smaller confidence intervals and standard errors than the baselines. It indicates that MFRS can provide stable learning results when employed on a real robot. In summary, it is verified that MFRS is able to handle the goal-conditioned tasks in the real-world scenario using the policy trained in the corresponding simulation environment, providing better performance and successfully improving the sample efficiency of the RL algorithm.

VI. CONCLUSION

In this paper, we propose a novel magnetic field-based reward shaping (MFRS) method for goal-conditioned RL tasks, where we consider the dynamic target and obstacles as permanent magnets and build our shaping reward function based on the intensity values of the magnet field generated by these magnets. MFRS is able to provide an explicit and informative optimization landscape for policy learning compared to the distance-based setting. To evaluate the validity and superiority of MFRS, we use CoppeliaSim to build a simulated 3-D robotic manipulation platform and generate a set of goal-conditioned tasks with various goal dynamics. Furthermore, we apply MFRS to a physical robot in the real-world environment with the policy trained in simulation. Experimental results both in simulation and on real robots verify that MFRS significantly improves the sample efficiency in goal-conditioned RL tasks with the dynamic target and obstacles compared to the relevant existing methods.

Our future work will focus on extending MFRS to the scenario with high-dimensional goals using the concept and properties of the magnetic field, and further generalizing to more diversified real-world tasks apart from the area of robotic control. Another significant direction would be incorporating some perception abilities with the decision-making RL, e.g., equipping the robot with additional sensors to obtain the pose of the dynamic target and obstacles, to fuse MFRS into an end-to-end integrated pipeline for more practical real-world validation of robotic systems.

APPENDIX
PROOF OF THE OPTIMAL POLICY INVARIANCE OF MFRS

Theorem 2: Let $M = (\mathcal{S}, \mathcal{G}, \mathcal{A}, \mathcal{T}, R, \gamma)$ be the original MDP with the environment reward R, and $M' = (\mathcal{S}, \mathcal{G}, \mathcal{A}, \mathcal{T}, R', \gamma)$ be the shaped MDP with the shaped reward $R' = R + F$, where the shaping reward function F satisfies (20) with the goal-conditioned potential function Φ initialized to zero. Let π^*_M and $\pi^*_{M'}$ be the optimal policies in M and M', respectively. Then, $\pi^*_{M'}$ is consistent with π^*_M.

Proof: According to UVFA [22], the optimal goal-conditioned Q-function in M should be equal to the expectation of long-term cumulative reward as

$$Q^*_M(s,a,g) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]. \tag{24}$$

Likewise, the optimal goal-conditioned Q-function in M' can be denoted as

$$\begin{aligned} Q^*_{M'}(s,a,g) &= E\left[\sum_{t=0}^{\infty} \gamma^t r'_t\right] \\ &= E\left[\sum_{t=0}^{\infty} \gamma^t (r_t + f_t)\right]. \end{aligned} \tag{25}$$

According to (20), we have

$$\begin{aligned} Q^*_{M'}(s,a,g) = E\Bigg[\sum_{t=0}^{\infty} & \gamma^t (r_t + \gamma \Phi_{t+1}(s_{t+1}, a_{t+1}, g) \\ & - \Phi_t(s_t, a_t, g))\Bigg] \\ = E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] &+ E\left[\sum_{t=1}^{\infty} \gamma^t \Phi_t(s_t, a_t, g)\right] \\ - E&\left[\sum_{t=0}^{\infty} \gamma^t \Phi_t(s_t, a_t, g)\right] \\ = E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] &- \Phi_0(s_0, a_0, g). \end{aligned} \tag{26}$$

Thus, we have $Q^*_{M'}(s,a,g) = Q^*_M(s,a,g) - \Phi_0(s_0, a_0, g)$, where $\Phi_0(s_0, a_0, g)$ denotes the initial value of the goal-conditioned potential function. The policy is obtained by maximizing the value of goal-conditioned Q-function, and hence the optimal policy in M' can be expressed as

$$\begin{aligned} \pi^*_{M'} &= \arg\max_{a \in A} Q^*_{M'}(s,a,g) \\ &= \arg\max_{a \in A} \left[Q^*_M(s,a,g) - \Phi_0(s_0, a_0, g)\right]. \end{aligned} \tag{27}$$

Therefore, if the value of Φ is initialized to zero, then we have $\pi^*_{M'} = \arg\max_{a \in A} Q^*_{M'}(s,a,g) = \pi^*_M$, which demonstrates that $\pi^*_{M'}$ is consistent with π^*_M in the goal-conditioned RL setting of our method. ■

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA, USA: MIT press, 2018.

[2] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[3] G. Tesauro and G. Galperin, "On-line policy improvement using monte-carlo search," in *Proc. Adv. Neural Inf. Proces. Syst.*, 1996.

[4] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3,

pp. 279–292, 1992.

[5] Z. Wang, C. Chen, H.-X. Li, D. Dong, and T.-J. Tarn, "Incremental reinforcement learning with prioritized sweeping for dynamic environments," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 2, pp. 621–632, 2019.

[6] C. Chen, D. Dong, H.-X. Li, J. Chu, and T.-J. Tarn, "Fidelity-based probabilistic Q-learning for control of quantum systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 920–933, 2013.

[7] B. Luo, D. Liu, T. Huang, and D. Wang, "Model-free optimal tracking control via critic-only Q-learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 10, pp. 2134–2144, 2016.

[8] Z. Zhang, D. Zhao, J. Gao, D. Wang, and Y. Dai, "FMRQ—A multiagent reinforcement learning algorithm for fully cooperative tasks," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1367–1379, 2016.

[9] Y. Zhu, D. Zhao, and X. Li, "Iterative adaptive dynamic programming for solving unknown nonlinear zero-sum game based on online data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 714–725, 2016.

[10] X. Xu, D. Hu, and X. Lu, "Kernel-based least squares policy iteration for reinforcement learning," *IEEE Trans. Neural Netw.*, vol. 18, no. 4, pp. 973–992, 2007.

[11] Y. Zheng, Z. Meng, J. Hao, Z. Zhang, T. Yang, and C. Fan, "A Bayesian policy reuse approach against non-stationary agents," in *Proc. Adv. Neural Inf. Proces. Syst.*, 2018.

[12] Y. Yu, S.-Y. Chen, Q. Da, and Z.-H. Zhou, "Reusable reinforcement learning via shallow trails," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2204–2215, 2018.

[13] Z. Wang, C. Chen, and D. Dong, "A Dirichlet process mixture of robust task models for scalable lifelong reinforcement learning," *IEEE Trans. Cybern.*, 2022. DOI: 10.1109/TCYB.2022.3170485

[14] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[15] O. Vinyals, I. Babuschkin, W. M. Czarnecki, *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[16] J. Pan, X. Wang, Y. Cheng, and Q. Yu, "Multisource transfer double DQN based on actor learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2227–2238, 2018.

[17] D. Silver, J. Schrittwieser, K. Simonyan, *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[18] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1329–1338.

[19] Z. Wang, C. Chen, and D. Dong, "Lifelong incremental reinforcement learning with online Bayesian inference," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 4003–4016, 2022.

[20] D. Li, D. Zhao, Q. Zhang, and Y. Chen, "Reinforcement learning and deep learning based lateral control for autonomous driving [application notes]," *IEEE Comput. Intell. Mag.*, vol. 14, no. 2, pp. 83–98, 2019.

[21] L. P. Kaelbling, "Learning to achieve goals," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 2, 1993, pp. 1094–8.

[22] T. Schaul, D. Horgan, K. Gregor, and D. Silver, "Universal value function approximators," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1312–1320.

[23] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," in *Proc. Adv. Neural Inf. Proces. Syst.*, 2017.

[24] O. Marom and B. Rosman, "Belief reward shaping in reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.

[25] M. Dann, F. Zambetta, and J. Thangarajah, "Deriving subgoals autonomously to accelerate learning in sparse reward domains," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 881–889.

[26] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proc. Int. Conf. Mach. Learn.*, vol. 99, 1999, pp. 278–287.

[27] E. Wiewiora, G. W. Cottrell, and C. Elkan, "Principled methods for advising reinforcement learning agents," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 792–799.

[28] S. M. Devlin and D. Kudenko, "Dynamic potential-based reward shaping," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2012, pp. 433–440.

[29] A. Harutyunyan, S. Devlin, P. Vrancx, and A. Nowé, "Expressing arbitrary reward functions as potential-based advice," in *Proc. AAAI Conf. Artif. Intell.*, vol. 29, no. 1, 2015.

[30] A. Trott, S. Zheng, C. Xiong, and R. Socher, "Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards," in *Proc. Adv. Neural Inf. Proces. Syst.*, 2019.

[31] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 387–395.

[32] N. Heess, G. Wayne, D. Silver, T. Lillicrap, T. Erez, and Y. Tassa, "Learning continuous control policies by stochastic value gradients," in *Proc. Adv. Neural Inf. Proces. Syst.*, 2015.

[33] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[34] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1587–1596.

[35] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[36] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, no. 3, pp. 229–256, 1992.

[37] M. Liu, M. Zhu, and W. Zhang, "Goal-conditioned reinforcement learning: Problems and solutions," arXiv preprint arXiv: 2201.08299, 2022.

[38] Y. Wu, G. Tucker, and O. Nachum, "The laplacian in RL: Learning representations with efficient approximations," arXiv preprint arXiv: 1810.04586, 2018.

[39] D. Ghosh, A. Gupta, and S. Levine, "Learning actionable representations with goal-conditioned policies," arXiv preprint arXiv: 1811.07819, 2018.

[40] D. Warde-Farley, T. Van de Wiele, T. Kulkarni, C. Ionescu, S. Hansen, and V. Mnih, "Unsupervised control through non-parametric discriminative rewards," arXiv preprint arXiv: 1811.11359, 2018.

[41] K. Hartikainen, X. Geng, T. Haarnoja, and S. Levine, "Dynamical distance learning for semi-supervised and unsupervised skill discovery," arXiv preprint arXiv: 1907.08225, 2019.

[42] I. Durugkar, M. Tec, S. Niekum, and P. Stone, "Adversarial intrinsic motivation for reinforcement learning," in *Proc. Adv. Neural Inf. Proces. Syst.*, 2021.

[43] J. Wu, Z. Huang, W. Huang, and C. Lv, "Prioritized experience-based reinforcement learning with human guidance for autonomous driving," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022. DOI: 10.1109/TNNLS.2022.3177685

[44] J. Wu, Z. Huang, Z. Hu, and C. Lv, "Toward human-in-the-loop AI: Enhancing deep reinforcement learning via real-time human guidance for autonomous driving," *Eng.*, 2022. DOI: 10.1016/j.eng.2022.05.017

[45] W. Bai, Q. Zhou, T. Li, and H. Li, "Adaptive reinforcement learning neural network control for uncertain nonlinear system with input saturation," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3433–3443, 2019.

[46] W. Bai, T. Li, Y. Long, and C. P. Chen, "Event-triggered multigradient recursive reinforcement learning tracking control for multiagent systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 1, pp. 366–379, 2021.

[47] Y. Ding, C. Florensa, P. Abbeel, and M. Phielipp, "Goal-conditioned imitation learning," in *Proc. Adv. Neural Inf. Proces. Syst.*, 2019.

[48] Y. Zhang, P. Abbeel, and L. Pinto, "Automatic curriculum learning through value disagreement," in *Proc. Adv. Neural Inf. Proces. Syst.*, 2020, pp. 7648–7659.

[49] R. McCarthy and S. J. Redmond, "Imaginary hindsight experience replay: Curious model-based learning for sparse reward tasks," arXiv preprint arXiv: 2110.02414, 2021.

[50] M. Fang, T. Zhou, Y. Du, L. Han, and Z. Zhang, “Curriculum-guided hindsight experience replay,” in *Proc. Adv. Neural Inf. Proces. Syst.*, 2019.

[51] S. Pitis, H. Chan, S. Zhao, B. Stadie, and J. Ba, “Maximum entropy gain exploration for long horizon multi-goal reinforcement learning,” in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7750–7761.

[52] Y. Kuang, A. I. Weinberg, G. Vogiatzis, and D. R. Faria, “Goal density-based hindsight experience prioritization for multi-goal robot manipulation reinforcement learning,” in *Proc. IEEE Int. Conf. Robot Human Interactive Commun.*, 2020, pp. 432–437.

[53] R. W. Pryor, *Multiphysics Modeling Using COMSOL®: A First Principles Approach*. Jones & Bartlett Publishers, 2009.

[54] R. E. Deakin, “3-D coordinate transformations,” *Surveying Land Inf. Syst.*, vol. 58, no. 4, pp. 223–234, 1998.

[55] H. B. Suay, T. Brys, M. E. Taylor, and S. Chernova, “Learning from demonstration for shaping through inverse reinforcement learning.” in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2016, pp. 429–437.

[56] G. A. Rummery and M. Niranjan, *On-line Q-learning Using Connectionist Systems*. Cambridge, UK: University of Cambridge, Department of Engineering, 1994, vol. 37.

[57] E. Rohmer, S. P. Singh, and M. Freese, “V-rep: A versatile and scalable robot simulation framework,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1321–1326.

[58] M. Plappert, M. Andrychowicz, A. Ray, *et al.*, “Multi-goal reinforcement learning: Challenging robotics environments and request for research,” arXiv preprint arXiv: 1802.09464, 2018.

[59] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “OpenAI Gym,” arXiv preprint arXiv: 1606.01540, 2016.

[60] M. R. Islam, M. A. Rahaman, M. Assad-uz Zaman, and M. Habibur, “Cartesian trajectory based control of dobot robot,” in *Proc. Int. Conf. Ind. Eng. Operations Manage.*, 2019.

[61] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” arXiv preprint arXiv: 1509.02971, 2015.

[62] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv: 1412.6980, 2014.

Hongyu Ding received the B.E. degree in mechanical engineering from the School of Mechanical and Power Engineering, East China University of Science and Technology, in 2021. He is currently pursuing the M.S. degree from the Department of Control Science and Intelligence Engineering, School of Management and Engineering, Nanjing University.

His current research interests include reinforcement learning, machine learning, and robotics.

Yuanze Tang received the B.E. degree in mechanical engineering, in 2022, from the School of Mechanical and Power Engineering, East China University of Science and Technology, where he is currently working toward the Ph.D. degree in digital science and engineering.

His current research interests include machine learning and intelligent life management for engineering equipment.

Qing Wu received the B.E. and M.E. degrees from the School of Mechanical and Power Engineering, East China University of Science and Technology, in 1995 and 2002. He is currently an Associate Professor in East China University of Science and Technology.

His recent research interests include machine learning, robotics, development and application of embedded system, computer measurement, and control of mechanical and electrical transmission.

Bo Wang (Member, IEEE) received the B.Sc. degree in software engineering from Southeast University, in 2007, and the M.Sc. and Ph.D. degrees from the Graduate School of Information, Production and Systems, Waseda University, Japan, in 2009 and 2012. He is currently an Associate Professor with the School of Management and Engineering, Nanjing University. He was a Research Assistant of the Global COE Program, Waseda University, Ministry of Education, Culture, Sports, Science and Technology, Japan. He was a Special Research Fellow of the Japan Society for the Promotion of Science (JSPS). He is a Committee Member of Chinese Association of Automation (CAA) Energy Internet Committee, and a Committee Member of Systems Engineering Society of JiangSu.

His research interests include, power system planning, renewable generation forecasting, data-driven decision-making, and artificial intelligence algorithms.

Chunlin Chen (Senior Member, IEEE) received the B.E. degree in automatic control and Ph.D. degree in control science and engineering from the University of Science and Technology of China, in 2001 and 2006, respectively. He is currently a Full Professor and the Vice Dean of School of Management and Engineering, Nanjing University. He was a Visiting Scholar at Princeton University, USA, from 2012 to 2013. He had visiting positions at the University of New South Wales, Australia, and the City University of Hong Kong, Hong Kong, China.

His recent research interests include reinforcement learning, mobile robotics, and quantum control. He is the Chair of Technical Committee on Quantum Cybernetics, IEEE Systems, Man and Cybernetics Society.

Zhi Wang (Member, IEEE) received the Ph.D. degree in machine learning from the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong, China, in 2019, and the B.E. degree in automation from Nanjing University, in 2015. He is currently an Associate Research Fellow with the Department of Control Science and Intelligence Engineering, School of Management and Engineering, Nanjing University. He holds visiting positions at the University of New South Wales, Australia, and the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences.

His current research interests include reinforcement learning, machine learning, and robotics. He served as the Associate Editor for special sessions of IEEE International Conference on Systems, Man, and Cybernetics 2021 and 2022, and IEEE International Conference on Networking, Sensing, and Control 2020.