2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Sample-Efficient Goal-Conditioned Reinforcement Learning via Predictive Information Bottleneck for Goal Representation Learning

Qiming Zou[1] and Einoshin Suzuki[2]

Abstract— **We propose Predictive Information bottleneck for Goal representation learning (PI-Goal), a self-supervised method for sample-efficient goal-conditioned reinforcement learning (RL). Goal-conditioned RL learns to reach commanded goals with reward signals. A goal could be given in a noisy or abstract form, and thus jeopardizes sample efficiency. Previous methods usually assume that the agent can map a state to an achievable goal. In this work, we consider a setting in which the goal space is unknown to the agent and the agent cannot recognize a goal in a specific state (referred to as a goal state) until the goal is commanded. Our PI-Goal learns a goal representation which contains only the predictive information of a goal state, i.e., the mutual information between a current state and a future state, and guarantees the optimality of the learned policy. Experimental results show that PI-Goal consistently outperforms the baseline methods in tasks with unknown goal spaces, e.g., object manipulation, object search, and embodied question answering.**

I. INTRODUCTION

A goal-oriented task, which requires an agent to reach desired goals, is ubiquitous not only in real-world applications such as object manipulation [1], visual navigation [2], and object search [3] but also in policy learning paradigms such as hierarchical RL [4], skill learning [5], and imitation learning [6]. Goal-conditioned RL [7] has the potential to train an agent so that it can accomplish diverse goal-oriented tasks without knowing a dynamic model of the environment [8].

A goal could be given in a noisy or abstract form, e.g., image or text goals. With such a form, two distinct goals can correspond to similar optimal goal-reaching policies. For example, in an embodied question answering task as shown in Fig. 1, the goal is to answer a given question with the gathered information at a specific location [9]. The questions which can be answered in adjacent time steps may be very different in texts. Learning to reach such a complex goal leads to a large sample requirement, compared to learning to reach a goal given in an informative form, e.g., the position of a goal, which necessitates learning a compact representation for a complex goal.

In this work, we for the first time consider a challenging yet practical setting: a goal-conditioned task with an unknown goal space. With this setting, a goal is commanded by a module unknown to the agent, e.g., humans. In an episode, the agent can determine whether it has achieved the commanded goal but cannot know the specific goals that might be commanded in future episodes. As a consequence, the mapping from a state to an achieved but uncommanded goal is unknown. For example, in an embodied question answering task, the agent can answer a given question based on an observed image, but it cannot know which question a human would ask for the observed image in the future. In the following, we show that previous methods cannot be directly applied to a task with an unknown goal space.

Previous works have extensively studied two ways to learn a goal representation: 1) learning the goal representation end-to-end with reward signals [10], [11] and 2) designing auxiliary self-supervised representation learning tasks [12], [13], [14]. The former approach is usually inefficient because the reward signal for reaching a goal is usually sparse [10], e.g., the reward signal is zero for a non-goal state. Hindsight Experience Replay (HER) makes the reward signal denser by replacing the commanded goal of a failed trajectory with another goal reached in this trajectory [11]. However, HER requires that the mapping between a state and an achieved goal is known, which does not hold for a task with an unknown goal space.

On the other hand, self-supervised representation learning creates denser learning signals via auxiliary learning tasks which can be classified as generative-based methods [12] or contrastive-based methods [15], [14]. A generative-based method usually optimizes a representation to reconstruct the goal, assuming that similar goal inputs correspond to similar optimal goal-reaching policies [12]. However, this assumption does not hold for noisy or abstract goals. Contrastive-based methods learn a representation which contains only information shared across multiple inputs or views of an input [15]. Previous contrastive-based methods often generate views via data augmentation, but it is difficult to select an augmentation method which guarantees that the shared information contains sufficient task-relevant information [13]. COGOAL [14] is a contrastive-based method that learns a goal representation by clustering the goals achieved in adjacent time steps in the latent space. When the goal space is unknown, the agent cannot assure whether a goal is within the goal space until it is commanded. As a consequence, the agent cannot evaluate the adjacency of multiple goals in a trajectory.

In short, previous goal representation learning methods assume that human knowledge [15], [16] or the goal space [11], [14] is available, which does not hold for a goal-conditioned

*This work was supported in part by China Scholarship Council (Grant No. 202008050300)

[1]Qiming Zou is with the Graduate School of Systems Life Sciences, Kyushu University, Fukuoka 819-0395, Japan zou.qiming.847@s.kyushu-u.ac.jp

[2]Einoshin Suzuki is with the Graduate School and Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan suzuki@inf.kyushu-u.ac.jp

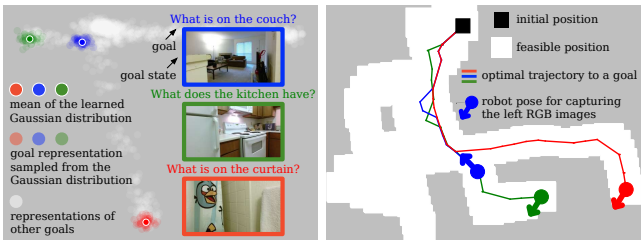


Fig. 1: **Example of an embodied question answering task modeled as a goal-conditioned task.** The goal state, the goal representation, the optimal goal-reaching trajectory, and the robot pose corresponding to the same goal share the same color. *Left:* the learned goal representations of PI-Goal and several pairs of (a goal, a goal state), i.e., (a question, an RGB image). *Right:* the optimal goal-reaching trajectories for three example goals and the poses at which the robot reaches the goals in an indoor environment. For different goals, the optimal policy similarity is reflected in the closeness between the learned goal representations, instead of the spatial or appearance similarity between the goals or the goal states.

task with an unknown goal space. In this work, we learn a compact goal representation for a task with an unknown goal space. With this compact representation, the policy avoids overfitting task-irrelevant details and instead generalizes its learned behavior to other goals which could be unseen.

Specifically, we learn a goal representation containing only predictive information, i.e., the mutual information between a current state and a future state, in the goal state and theoretically prove that it is sufficient for learning the optimal goal-reaching policy. The assumption is that a goal can be reached at a goal state in a stationary state space. Based on the assumption, we have 1) For an agent, reaching a goal is equivalent to reaching the goal state; 2) It is relatively easy to evaluate the state predictive information with an arbitrary policy. According to the above two facts, it is reasonable to leverage the state predictive information to embed task-relevant information in the goal representation, thus overcoming the challenge posed by the unknown goal space.

To summarize, our contributions are two-fold: 1) We propose a general and theoretically inspired goal representation learning method for sample-efficient goal-conditioned RL; 2) We demonstrate the performance of our method in three tasks with unknown goal spaces.

II. Related Works

A. Self-Supervised State Representation Learning

Several works utilize an auto-encoding model to learn a state representation with an auxiliary state reconstruction task in which the representation and a decoder network are optimized jointly to reconstruct the state [17], [12]. However, a reconstruction-based method encodes all the information contained in the raw input, which tend to make a policy overfit to task-irrelevant information. Data augmentation is a common choice to avoid overfitting by randomizing task-irrelevant information in a state [18], e.g., color randomization and random rotation. Another line of works either augment states along with RL [19] or design auxiliary contrastive learning tasks to learn the state representation independently [15]. However, these methods require human knowledge to select the type of data augmentation which does not randomize task-relevant information [13]. Recent works learn a state representation that maximizes the predictive information for standard RL [20], [21]. This state representation has been utilized as subgoals to guide the agent towards the commanded goal and improves the sample efficiency of goal-conditioned RL [22], [23]. Our PI-Goal learns a goal representation based on this state representation and is complementary to the above methods.

B. Self-Supervised Goal Representation Learning

Reconstruction-based methods are extensively utilized in goal representation learning [1], [24]. Particularly, disentangled goal representation leads to better sample efficiency [25], [26]. However, these methods either include task-irrelevant information [1], [24], [26] or rely on human knowledge to extract specific attributes from the goal input [25]. Recent works [27], [28] have sought to adopt the information bottleneck [29] for goal representation learning. However, these works ensure task-relevant information either through sparse rewards [27] or the optimal action at each state [28]. There are works which propose to group temporally adjacent goals, i.e., goals which can be reached in a small time window, in a latent space [14], [30]. These methods require recognizing every goal in each state, i.e., they assume that a state-goal mapping function is available. However, in a task with an unknown goal space, the agent cannot recognize a goal until the goal is commanded, and thus cannot directly evaluate the adjacency between multiple goals.

III. Preliminary

A. Goal-Conditioned Reinforcement Learning

Goal-conditioned RL is typically formalized as a Markov Decision Process (MDP) [31]. An MDP is defined as a tuple $<\mathcal{S}, \mathcal{A}, \mathcal{G}, P, r, \gamma>$, where $\mathcal{S} \subseteq \mathbb{R}^{d_{\mathbf{s}}}$ is the state space, $\mathcal{G} \subseteq \mathbb{R}^{d_{\mathbf{g}}}$ is the goal space, and P denotes the state transition function. We assume an MDP with a discrete action space, i.e., action space $\mathcal{A} \subseteq \mathbb{Z}^+$, and $|\mathcal{A}| = d_a$.

There is an unknown mapping from state $\mathbf{s}$ to goal $\mathbf{g}$, i.e., $\mathbf{g} = m(\mathbf{s})$. The reward function r is defined as

$$r(\mathbf{s}, \mathbf{g}) := \begin{cases} 1 & \text{if } \mathbf{g} = m(\mathbf{s}), \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Interaction record $\tau = \{(\mathbf{s_t}, a_t, r_t, \mathbf{g})\}_{t=0,1,\cdots,T_\tau-1}$ is called a trajectory, where T_τ is the length of τ. A trajectory ends when commanded goal $\mathbf{g} \in \mathcal{G}$ is reached.

The objective is to find optimal policy π^* that maximizes its expected discounted sum of rewards $V^\pi(\mathbf{s}, \mathbf{g})$. With

reward function $r(\mathbf{s}, \mathbf{g})$ defined in Eq. (1). $V^{\pi}(\mathbf{s}, \mathbf{g})$ takes the form

$$V^{\pi}(\mathbf{s}, \mathbf{g}) \;=\; \mathbb{E}_{\tau}\left[\gamma^{T_\tau} \middle| \pi, \mathbf{g}, \mathbf{s}_0 = \mathbf{s}\right], \tag{2}$$

where γ represents a discount factor $(0 < \gamma < 1)$.

B. Representation for Goal-Conditioned RL

An encoder is a stochastic mapping from input space $\mathcal{X}$, i.e., $\mathcal{S}$ or $\mathcal{G}$, to a probability distribution over the elements of representation space $\mathcal{Z} \subseteq \mathbb{R}^{d_{\mathbf{z}}}$, where $d_{\mathbf{z}}$ is the dimension of the representation. In this work, the probability distribution is diagonal Gaussian distribution $\mathcal{N}(\mu, \sigma)$ with mean $\mu \in \mathbb{R}^{d_{\mathbf{z}}}$ and standard variance $\sigma \in \mathbb{R}^{d_{\mathbf{z}}}$. Representation $\mathbf{z}$ is sampled from $\mathcal{N}(\mu, \sigma)$ with the reparameterization trick [32].

Parameters $(\mu_{\mathbf{s}}, \sigma_{\mathbf{s}})$ for state encoder κ and parameters $(\mu_{\mathbf{g}}, \sigma_{\mathbf{g}})$ for goal encoder ϕ are generated by two separate neural networks. The neural network takes an input, i.e., $\mathbf{s}$ or $\mathbf{g}$, and outputs a vector, i.e., $[\mu_{\mathbf{s}}, \sigma_{\mathbf{s}}]$ or $[\mu_{\mathbf{g}}, \sigma_{\mathbf{g}}]$. The output vector is split into the mean and the variance parameters. Note that PI-Goal needs auxiliary state encoder ψ for extracting the predictive information contained in the state input.

C. Predictive Information Bottleneck

MI $I(\mathbf{x}; \mathbf{y})$ quantifies the "amount of information" obtained about random variable $\mathbf{x}$ by observing random variable $\mathbf{y}$ [29].

The predictive information [33] is the MI between the past and the future, $I(\mathbf{z}_{\mathbf{s}}; \mathbf{z}_{\mathbf{s}^+})$, where $\mathbf{z}_{\mathbf{s}}$ and $\mathbf{z}_{\mathbf{s}^+}$ are representations of current state $\mathbf{s}$ and future state $\mathbf{s}^+$, respectively. Note that the state representation follows a Gaussian distribution generated by the state encoder. The predictive information bottleneck (PIB) learns a representation which only contains information for predicting the representation of the next state [20]. Specifically, the PIB problem for $(\mathbf{z}_{\mathbf{s}}, \mathbf{z}_{\mathbf{s}^+})$ is given as

$$\min_{\psi} \quad \beta \overbrace{I(\mathbf{z}_{\mathbf{s}}; \mathbf{s})}^{\text{state infomin}} - \overbrace{I(\mathbf{z}_{\mathbf{s}}; \mathbf{z}_{\mathbf{s}^+})}^{\text{state infomax}}, \tag{3}$$

where $\mathbf{z}_{\mathbf{s}} = \psi(\mathbf{s})$ and $\mathbf{z}_{\mathbf{s}^+} = \psi(\mathbf{s}^+)$. The degree of compression increases as β goes from 0 to 1.

To optimize the loss function with MI terms end-to-end, we approximate the MI via variational bounds. For MI minimization, we adopt an upper bound derived in [34], which is a KL-divergence between a representation and a standard Normal distribution. For MI maximization, the MI can be estimated by maximizing infoNCE [35] which is a sample-based differentiable MI lower bound.

IV. Predictive Information Bottleneck for Goal Representation Learning (PI-Goal)

A. Sufficient Goal Representation

Intuitively, given state $\mathbf{s}$, a value estimation function which takes sufficient goal representation $\mathbf{z}$ as input must be able to predict optimal cumulative reward $V^*_{\mathbf{s},\mathbf{g}}$ at least as accurately as if it has access to original goal $\mathbf{g}$. Optimal cumulative reward $V^*_{\mathbf{s},\mathbf{g}}$ is defined as $V^*_{\mathbf{s},\mathbf{g}} = V^{\pi^*}(\mathbf{s}, \mathbf{g})$. We formalize the above intuition as follows.

Definition 1 (Sufficiency of Goal Representation): Combined with state $\mathbf{s}$, representation $\mathbf{z} = \phi(\mathbf{g})$ is sufficient for predicting $V^*_{\mathbf{s},\mathbf{g}}$ if and only if $I(\mathbf{z}, \mathbf{s}; V^*_{\mathbf{s},\mathbf{g}}) = I(\mathbf{g}, \mathbf{s}; V^*_{\mathbf{s},\mathbf{g}})$. In other words, $(\mathbf{s}, \mathbf{z})$ encodes all the information contained in $(\mathbf{s}, \mathbf{g})$ which is useful for predicting $V^*_{\mathbf{s},\mathbf{g}}$.

B. Assumption

To learn a sufficient goal representation, we set the following assumptions.

Definition 2 (Mutual Redundancy [36]): Two random variables $\mathbf{v_1}$ and $\mathbf{v_2}$ are mutually redundant for target variable $\mathbf{y}$ if and only if $I(\mathbf{y}; \mathbf{v_1} \mid \mathbf{v_2}) = I(\mathbf{y}; \mathbf{v_2} \mid \mathbf{v_1}) = 0$.

Assumption 1: Combined with state $\mathbf{s}$, goal state $\mathbf{s_g}$ and state $\mathbf{s_g^+}$ (the next state of $\mathbf{s_g}$ in a trajectory) are mutually redundant for optimal cumulative reward $V^*_{\mathbf{s},\mathbf{g}}$.

Intuitively, observing $\mathbf{s_g}$ introduces no new useful information for predicting $V^*_{\mathbf{s},\mathbf{g}}$ given that $\mathbf{s_g^+}$ is observed. We believe this assumption is reasonable, since $V^*_{\mathbf{s},\mathbf{g}} \approx V^*_{\mathbf{s},\mathbf{g}^+}$ in a goal-conditioned MDP, where $\mathbf{g}^+$ is reached in state $\mathbf{s_g^+}$. Specifically, since $\mathbf{g}^+$ can be reached by taking one step further after reaching $\mathbf{g}$, we have $\left|V^*_{\mathbf{s},\mathbf{g}} - V^*_{\mathbf{s},\mathbf{g}^+}\right| = \gamma^{T_\tau} - \gamma^{T_\tau+1} \leq 1 - \gamma$, where γ is usually close to 1.

Assumption 2: Combined with state $\mathbf{s}$, goal state $\mathbf{s_g}$ and goal $\mathbf{g}$ share the same amount of information for predicting $V^*_{\mathbf{s},\mathbf{g}}$, i.e., $I(\mathbf{s_g}, \mathbf{s}; V^*_{\mathbf{s},\mathbf{g}}) = I(\mathbf{g}, \mathbf{s}; V^*_{\mathbf{s},\mathbf{g}})$.

Intuitively, we assume that reaching a goal is equivalent to reaching the state at which the condition of reaching the goal is satisfied. This assumption holds true in a wide range of goal-conditioned tasks. For example, in an embodied question answering task as shown in Fig. 1, there is an image which can provide sufficient information for answering the given question. Another example is the object search task, searching for a target object is equivalent to searching for a location at which the target object can be found.

C. Learning Sufficient Goal Representation

In this section, we prove that learning a sufficient goal representation is equivalent to maximizing $I(\mathbf{z}; \mathbf{z}_{\mathbf{s_g}})$, $\mathbf{z} = \phi(\mathbf{g})$, $\mathbf{z}_{\mathbf{s_g}} = \psi^*(\mathbf{s_g})$. State encoder ψ^* is obtained by optimizing the loss function in Eq. (3).

In Theorem 1, we show that $\mathbf{z}_{\mathbf{s_g}}$ shares the same amount of information for predicting $V^*_{\mathbf{s},\mathbf{g}}$ as goal $\mathbf{g}$. To prove Theorem 1, we introduce the following lemma.

Lemma 1 ([36]): Let $\mathbf{v_1}$ and $\mathbf{v_2}$ be mutually redundant views for $\mathbf{y}$. Let $\mathbf{z_1}$ be a representation of $\mathbf{v_1}$ that is sufficient for $\mathbf{v_2}$, then $I(\mathbf{z_1}; \mathbf{y}) = I(\mathbf{v_1}, \mathbf{v_2}; \mathbf{y})$.

Proof: See Corollary 1 in [36]. ■

Theorem 1: Let $\mathbf{s_g}$ be the state at which the agent reaches goal $\mathbf{g}$. Suppose representation $\mathbf{z}_{\mathbf{s_g}}$ encodes the predictive information of $\mathbf{s_g}$, i.e., $I(\mathbf{z}_{\mathbf{s_g}}; \mathbf{z}_{\mathbf{s_g^+}}) = I(\mathbf{s_g}; \mathbf{s_g^+})$. We have that $(\mathbf{s}, \mathbf{g})$ and $(\mathbf{s}, \mathbf{z}_{\mathbf{s_g}})$ are mutually redundant for $V^*_{\mathbf{s},\mathbf{g}}$.

Proof: With Assumption 1, we regard $(\mathbf{s_g}, \mathbf{s})$, $(\mathbf{s_g^+}, \mathbf{s})$, $(\mathbf{z}_{\mathbf{s_g}}, \mathbf{s})$, and $V^*_{\mathbf{s},\mathbf{g}}$ as $\mathbf{v_1}$, $\mathbf{v_2}$, $\mathbf{z}$, and $\mathbf{y}$ in Lemma 1,

respectively. We have

$$I(\mathbf{z}_{\mathbf{s_g}}, \mathbf{s}; V^*_{\mathbf{s,g}}) \overset{\text{Lemma 1}}{=} I(\mathbf{s_g}, \mathbf{s}^+_{\mathbf{g}}, \mathbf{s}; V^*_{\mathbf{s,g}}) \geq I(\mathbf{s_g}, \mathbf{s}; V^*_{\mathbf{s,g}}) \overset{\text{Assumption 2}}{=} I(\mathbf{g}, \mathbf{s}; V^*_{\mathbf{s,g}}).$$

Data Processing Inequality (DPI) [37] states that processing a random variable cannot increase information. According to DPI, we have

$$I(\mathbf{z}_{\mathbf{s_g}}, \mathbf{s}; V^*_{\mathbf{s,g}}) \overset{\text{DPI}}{\leq} I(\mathbf{s_g}, \mathbf{s}; V^*_{\mathbf{s,g}}) \overset{\text{Assumption 2}}{=} I(\mathbf{g}, \mathbf{s}; V^*_{\mathbf{s,g}}).$$

By combining the above two equations, we have $I(\mathbf{z}_{\mathbf{s_g}}, \mathbf{s}; V^*_{\mathbf{s,g}}) = I(\mathbf{g}, \mathbf{s}; V^*_{\mathbf{s,g}})$.

Finally, we prove that $(\mathbf{z}_{\mathbf{s_g}}, \mathbf{s})$ and $(\mathbf{g}, \mathbf{s})$ are mutually redundant for $V^*_{\mathbf{s,g}}$. Since $(\mathbf{g}, \mathbf{s})$ contains all the task-relevant information, we have $I(V^*_{\mathbf{s,g}}; \mathbf{z}_{\mathbf{s_g}}, \mathbf{s} | \mathbf{g}, \mathbf{s}) = 0$. On the other hand, we have

$$I(V^*_{\mathbf{s,g}}; \mathbf{g}, \mathbf{s} | \mathbf{z}_{\mathbf{s_g}}, \mathbf{s}) = I(\mathbf{g}, \mathbf{s}; V^*_{\mathbf{s,g}}) - I(V^*_{\mathbf{s,g}}; \mathbf{g}, \mathbf{s}; \mathbf{z}_{\mathbf{s_g}}, \mathbf{s})$$
$$= \left[I(\mathbf{g}, \mathbf{s}; V^*_{\mathbf{s,g}}) - I(\mathbf{z}_{\mathbf{s_g}}, \mathbf{s}; V^*_{\mathbf{s,g}})\right] - I(V^*_{\mathbf{s,g}}; \mathbf{z}_{\mathbf{s_g}}, \mathbf{s} | \mathbf{g}, \mathbf{s}) = 0$$

Therefore, $I(V^*_{\mathbf{s,g}}; \mathbf{g}, \mathbf{s} | \mathbf{z}_{\mathbf{s_g}}, \mathbf{s}) = I(V^*_{\mathbf{s,g}}; \mathbf{z}_{\mathbf{s_g}}, \mathbf{s} | \mathbf{g}, \mathbf{s}) = 0$, which proves the theorem. ■

Based on Definition 1 and Theorem 1, we have the following Theorem.

Theorem 2: Let $\mathbf{z} = \phi(\mathbf{g})$ and $\mathbf{s_g}$ be the state at which the agent reaches goal $\mathbf{g}$. Suppose representation $\mathbf{z}_{\mathbf{s_g}}$ encodes the predictive information of $\mathbf{s_g}$. If $I(\mathbf{z}; \mathbf{z}_{\mathbf{s_g}})$ is maximized, combined with state $\mathbf{s}$, $\mathbf{z}$ is sufficient for $V^*_{\mathbf{s,g}}$.

Proof: Representation $\mathbf{z}_{\mathbf{s_g}}$ encodes the predictive information of $\mathbf{s_g}$. Based on Theorem 1, $(\mathbf{s}, \mathbf{g})$ and $(\mathbf{s}, \mathbf{z}_{\mathbf{s_g}})$ are mutually redundant for $V^*_{\mathbf{s,g}}$.

We regard $(\mathbf{g}, \mathbf{s})$, $(\mathbf{z}_{\mathbf{s_g}}, \mathbf{s})$, $(\mathbf{z}, \mathbf{s})$, and $V^*_{\mathbf{s,g}}$ as $\mathbf{v_1}$, $\mathbf{v_2}$, $\mathbf{z}$, and $\mathbf{y}$ in Lemma 1. We have

$$I(\mathbf{z}, \mathbf{s}; V^*_{\mathbf{s,g}}) \overset{\text{Lemma 1}}{=} I(\mathbf{z}_{\mathbf{s_g}}, \mathbf{g}, \mathbf{s}; V^*_{\mathbf{s,g}}) \overset{\text{DPI}}{\geq} I(\mathbf{g}, \mathbf{s}; V^*_{\mathbf{s,g}}).$$

We have $I(\mathbf{z}, \mathbf{s}; V^*_{\mathbf{s,g}}) \overset{\text{DPI}}{\leq} I(\mathbf{g}, \mathbf{s}; V^*_{\mathbf{s,g}})$. Finally, $I(\mathbf{z}, \mathbf{s}; V^*_{\mathbf{s,g}}) = I(\mathbf{g}, \mathbf{s}; V^*_{\mathbf{s,g}})$, which proves the theorem. ■

Based on Theorem 2 and information bottleneck [29], we propose a loss function,

$$\min_{\phi} \quad \beta \overbrace{I(\mathbf{z}; \mathbf{g})}^{\text{goal infomin}} - \overbrace{I(\mathbf{z}; \mathbf{z}_{\mathbf{s_g}})}^{\text{goal infomax}}, \tag{4}$$

where $\mathbf{s_g}$ is the state at which $\mathbf{g}$ can be reached, $\mathbf{z} = \phi(\mathbf{g})$, and $\mathbf{z}_{\mathbf{s_g}} = \psi^*(\mathbf{s_g})$.

D. Practical Algorithm

In this section, we propose a practical algorithm for goal-conditioned RL with PI-Goal. In goal-conditioned MDP $\mathcal{M}$, the agent 1) collects trajectory τ with policy π given commanded goal $\mathbf{g}$; 2) samples temporally adjacent state pairs $(\mathbf{s}, \mathbf{s}^+)$ from τ; 3) learns a state representation containing only predictive information via updating Eq. (3); 4) learns a compact goal representation via updating Eq. (4) if $\mathbf{s_g} \in \tau$; 5) updates goal encoder ϕ, state encoder κ, and policy π jointly with an off-the-shelf RL algorithm, e.g., DQN; 6) goes back to step 1) unless reaching maximum training episode number N; 7) finishes training. The details are shown in Algorithm 1. The sample efficiency of PI-Goal can be further enhanced by pre-training state encoder ψ if a pre-collected trajectory dataset is available. It is important to note that in step 4), the agent can only identify the goal states of the commanded goals in τ. This is because the goal space is unknown to the agent. Therefore, the agent cannot determine whether other achieved goals are within the goal space.

Algorithm 1 Goal-Conditioned RL with PI-Goal

Input: Goal encoder ϕ, state encoder ψ, state encoder κ, policy π, environment $\mathcal{M}$, number N of the RL episodes, maximum number T of the episode time steps
Output: Fully trained ϕ, ψ, κ and π

1: **for** episodes $= 0, 1, \cdots, N-1$ **do**
2: Uniformly sample goal $\mathbf{g} \in \mathcal{G}$ and initial state $\mathbf{s}_0 \in \mathcal{S}$.
3: Roll-out π, producing trajectory τ.
4: Randomly sample state pair $(\mathbf{s}, \mathbf{s}^+) = (\mathbf{s_t}, \mathbf{s_{t+1}})$, $\mathbf{s_t} \in \tau$, $\mathbf{s_{t+1}} \in \tau$.
5: Update ψ via Eq. (3).
6: **if** $\mathbf{s_g} \in \tau$ **then**
7: Update ϕ via Eq. (4).
8: **end if**
9: Update ϕ, π, and κ with τ using an off-the-shelf RL algorithm.
10: **end for**

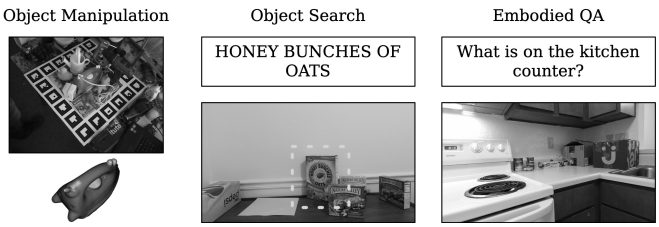


Fig. 2: Example of a goal (shown in the top row) and a goal state (shown in the bottom row) in our benchmark environments. A column corresponds to an environment.

V. Experiments

In this Section, we aim at answering two questions: 1) Does PI-Goal improve the sample efficiency of an RL agent? 2) What are the gains from PI-Goal?

A. Benchmark Environments

In an environment, an RL agent is trained to reach a set of goals starting from a random initial state. Examples of a goal and the corresponding goal state in each environment are shown in Fig. 2. We consider the following environments in the experiments.

Object Manipulation: An object can be rotated around the pitch, yaw, and roll axes. The agent takes a grey image of an object as observation and manipulates the object to a pose given in another grey image of the scene within 100 time steps. The agent learns to manipulate the object, i.e., the iron shown in Fig. 2, for reaching total of 500 poses in different

scenes. The 3D object model and the scene images are from the dataset [38].

Object Search [39]: An automobile wheeled robot searches for target objects within 100 time steps by taking six kinds of actions: forward, backward, left, right, clockwise rotation, and counter-clockwise rotation. The size of the goal space is 24. The robot receives a grey image as observation and a target object as a goal (randomly sampled from 24 object categories in each episode). The goal is given in text, e.g., "coca cola glass bottle", which is transformed to a 384 dimensional vector with SBERT [40].

Embodied Question Answering (Embodied QA): An automobile wheeled robot (the same robot as in the Object Search task) navigates to a specific location to answer a question (considered as the goal). The size of the goal space is 138. We construct a labeled dataset based on a realistic visual navigation dataset [39]. Specifically, we generate question-answer pairs at each state (a grey image) with two steps: 1) Generating an image caption via One For All [41]; 2) Generating question-answer pairs from the image caption [42]. The question inquires about specific details in the image caption and the answer is an extract of that information from the caption.

B. Baseline Methods

We compare PI-Goal with the following baseline methods:

VAE [1]: A method which optimizes a goal representation to reconstruct the goal input.

RAD [16] An RL algorithm which augments the input. We augment the goal input via random translation for an image goal and random amplitude scaling for a text goal.

CURL [15] An RL algorithm combined with contrastive learning on augmented views of the goal input.

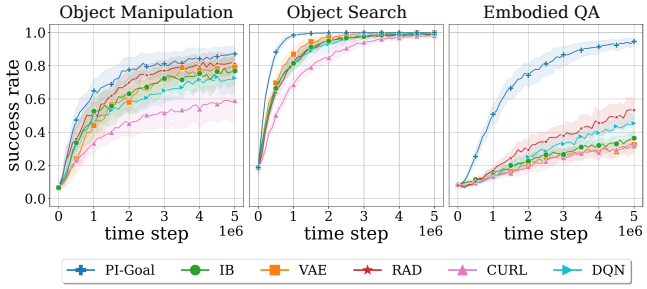
IB [27] A method which minimizes the MI between the goal representation and the goal input.

We combine the above methods with DQN [43] which is a standard value-based algorithm for discrete action tasks. Since the agent cannot recognize other goals except the commanded goals in a trajectory, we cannot use a trajectory relabelling based method, e.g., HER [11] and COGOAL [14]. To evaluate the robustness of each method, we repeat the experiments for 10 random seeds.

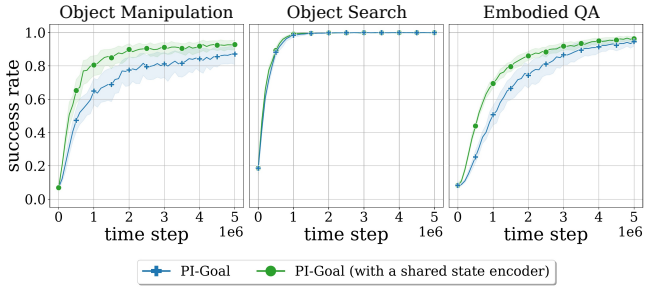
C. Comparison Results

As shown in Fig. 3a, PI-Goal significantly improves the sample-efficiency of DQN in the three tasks compared to the baseline methods. In Table I, we compare the performance given at a fixed number of environment interactions, i.e., 1 million (1M) and 5 million (5M). PI-Goal consistently outperforms the baseline methods in both the sample-efficiency (1M) and the asymptotic performance (5M). All methods achieve comparable performances in Object Search task at 5M time step, since this task contains a smaller number of goals, compared to the other tasks.

We evaluate the effect of using ψ for both PI-Goal and policy π.



(a) Learning curves of PI-Goal and baseline methods.



(b) Learning curves of PI-Goal and PI-Goal (with a shared state encoder). The solid lines and shaded regions represent the mean and the standard deviation over 10 random seeds, respectively.

Fig. 3: Learning curves of goal representation methods combined with DQN on the benchmark environments.

TABLE I: Success rate at 1M and 5M environment steps. In a task, a bold figure represents the overall best, an underlining number represents the second best.

1M time step	**PI-Goal**	IB	VAE	RAD	CURL	DQN
Object Manipulation	**0.65**	0.53	0.43	0.52	0.33	0.46
Object Search	**0.98**	0.82	0.87	0.82	0.69	0.80
Embodied QA	**0.49**	0.16	0.14	0.17	0.13	0.13
5M time step						
Object Manipulation	**0.87**	0.78	0.78	0.80	0.58	0.76
Object Search	**1.0**	0.99	**1.0**	0.99	0.98	0.99
Embodied QA	**0.94**	0.37	0.31	0.53	0.32	0.44

PI-Goal (with a shared state encoder): A variant which utilizes predictive information in both goal and state representation learning by replacing κ with ψ for policy π. Previous works [20], [21] have empirically proved the effectiveness of the predictive information in state representation learning. Note that PI-Goal (with a shared state encoder) further improves the performance of PI-Goal in all three benchmarks, as shown in Fig. 3b.

D. Visualizations of the Learned Goal Representation

To understand how PI-Goal achieves high sample efficiency, we visualize its learned goal representations for Object Manipulation and Object Search via Principal Component Analysis (PCA) [44] in Fig. 4. The visualization for Embodied QA is shown in Fig. 1. According to Assumption 1, the goals (the circles with the same color), which correspond to closely located goal states, tend to share similar optimal goal-reaching policies. Therefore, ideally, the circles with the same color should be grouped into a cluster in the visualization.

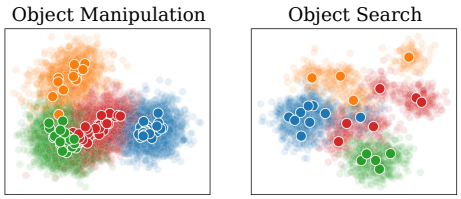


Fig. 4: Learned goal representations of PI-Goal. For a goal, a filled circle is the mean of a Gaussian distribution generated by PI-Goal, and a faded circle is a representation drawn from the distribution.

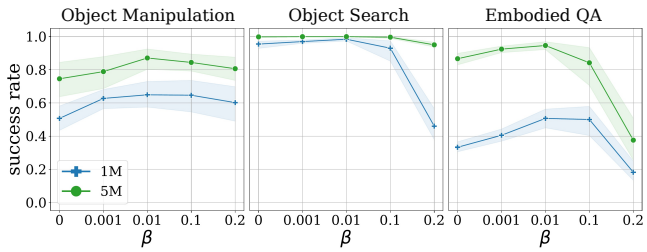


Fig. 5: Success rate in terms of β at time steps 1M and 5M.

According to the visualization results in Fig. 1 and Fig. 4, we find that PI-Goal learns a goal representation with two properties: 1) PI-Goal groups the representations of similar goals; 2) PI-Goal learns goal representations which mutually overlap each other to a certain degree. These two properties indicate that the majority of the information contained in the learned representation is task-relevant. Utilizing the learned representation as an input, the policy will not overfit task-irrelevant information. This characteristic leads to improved sample efficiency, as the learned behavior for a goal can be applied to other training goals which are reached by similar optimal policies. Alternatively, the learned behavior can be generalized to reach similar but unseen goals, although the generalizability is not the primary focus of the work.

E. Hyper-Parameter Analysis and Ablation Study

Our method has only one hyper-parameter to set, i.e., the weight of infomin term β. We empirically studied the effect of different β. As shown in Fig. 5, the success rate increases between $\beta = 0$ and $\beta = 0.01$, followed by a drop. Small and large β values lead to inferior performance for different reasons. In the case of small β, the infomax term dominates the infomin term, resulting in goal representations that are more specific to each goal and have reduced overlap. The reduced overlap hinders the learning process as the policy learns to reach each goal in a more independent manner. When β is large, the infomin term dominates the infomax term, resulting in goal representations that are close to a standard Gaussian distribution. This situation leads to a loss of task-relevant information in the goal representation, making it harder for the policy to differentiate goals and requiring larger samples.

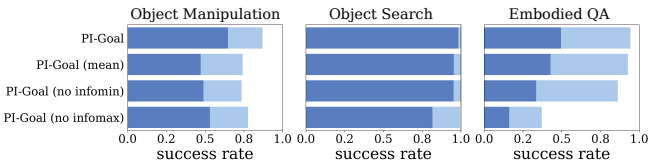


Fig. 6: Ablation study of the infomin/infomax term and the representation uncertainty. We compare the training performance of PI-Goal against a number of variants at 1M (dark blue bar) and 5M (light blue bar) time steps, respectively. The light and dark blue bars overlap each other.

PI-Goal generates a Gaussian distribution for a goal and samples a point from the distribution as a goal representation. To investigate the effectiveness of the uncertainty of the representation, we compare PI-Goal with PI-Goal (mean).

PI-Goal (mean): A variant which utilizes the mean of the Gaussian distribution as the goal representation.

PI-Goal has two components, i.e., the infomin term and the infomax term, shown in Eq. (3) and Eq. (4), respectively. We compare PI-Goal with the following variants to investigate the effectiveness of each component.

PI-Goal (no infomin): A variant which learns a goal representation by removing the state and goal infomin terms.

PI-Goal (no infomax): A variant which learns a goal representation by removing the state and goal infomax terms.

The success rate of PI-Goal and its variants is shown in Fig. 6. Since the uncertainty of the representation increases the overlap among similar goals as shown in Fig. 1 and Fig. 4, PI-Goal consistently outperforms PI-Goal (mean). The comparison results between PI-Goal and its variants without the infomin term support the common intuition that reducing irrelevant information improves the sample-efficiency. In addition, PI-Goal outperforms its variants without infomax terms by a large margin.

VI. Conclusion

For a goal-conditioned task with an unknown goal space, we proposed PI-Goal which utilizes predictive information in a goal state to learn a compact goal representation. Since the predictive information in a state can be easily evaluated in a trajectory and contains sufficient task-relevant information, PI-Goal achieves high sample efficiency in three tasks, i.e., object manipulation, object search, and embodied question answering. In summary, we believe that PI-Goal broadens the scope of goal-conditioned RL.

An exciting direction for future research is to consider settings where the state space is non-stationary. In this setting, the state reaching a goal would be non-stationary, which poses the difficulty of inferring the goal representation with PI-Goal. To overcome the difficulty, further training (or fine-tuning) of both the goal encoder and the policy would be necessary.

References

[1] A. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine, "Visual Reinforcement Learning with Imagined Goals," in *Proc. NeurIPS*, 2018, pp. 9209–9220.

[2] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-Driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning," in *Proc. ICRA*, 2017, pp. 3357–3364.

[3] A. Mousavian, A. Toshev, M. Fiser, J. Kosecká, A. Wahid, and J. Davidson, "Visual Representations for Semantic Target Driven Navigation," in *Proc. ICRA*, 2019, pp. 8846–8852.

[4] O. Nachum, S. S. Gu, H. Lee, and S. Levine, "Data-Efficient Hierarchical Reinforcement Learning," in *Proc. NeurIPS*, 2018, pp. 3307–3317.

[5] V. Campos, A. Trott, C. Xiong, R. Socher, X. Giró-i-Nieto, and J. Torres, "Explore, Discover and Learn: Unsupervised Discovery of State-Covering Skills," in *Proc. ICML*, vol. 119, 2020, pp. 1317–1327.

[6] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, "Relay Policy Learning: Solving Long-Horizon Tasks via Imitation and Reinforcement Learning," in *Proc. CoRL*, 2019, pp. 1025–1037.

[7] L. P. Kaelbling, "Learning to Achieve Goals," in *Proc. IJCAI*, 1993, pp. 1094–1099.

[8] T. Schaul, D. Horgan, K. Gregor, and D. Silver, "Universal Value Function Approximators," in *Proc. ICML*, 2015, pp. 1312–1320.

[9] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *Proc. CVPR*, 2018, pp. 1–10.

[10] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking Deep Reinforcement Learning for Continuous Control," in *Proc. ICML*, 2016, pp. 1329–1338.

[11] M. Andrychowicz, D. Crow, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight Experience Replay," in *Proc. NeurIPS*, 2017, pp. 5048–5058.

[12] S. Lange and M. A. Riedmiller, "Deep Auto-Encoder Neural Networks in Reinforcement Learning," in *Proc. IJCNN*, 2010, pp. 1–8.

[13] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What Makes for Good Views for Contrastive Learning," *Proc. NeuriPS*, 2020.

[14] Q. Zou and E. Suzuki, "Contrastive Goal Grouping for Policy Generalization in Goal-Conditioned Reinforcement Learning," in *Proc. ICONIP*, 2021, pp. 240–253.

[15] M. Laskin, A. Srinivas, and P. Abbeel, "CURL: Contrastive Unsupervised Representations for Reinforcement Learning," in *Proc. ICML*, 2020, pp. 5639–5650.

[16] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, "Reinforcement Learning with Augmented Data," in *Proc. NeurIPS*, 2020.

[17] D. Yarats, A. Zhang, I. Kostrikov, B. Amos, J. Pineau, and R. Fergus, "Improving Sample Efficiency in Model-Free Reinforcement Learning from Images," in *Proc. AAAI*, 2021, pp. 10 674–10 681.

[18] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, p. 60, 2019.

[19] I. Kostrikov, D. Yarats, and R. Fergus, "Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels," in *Proc. ICLR*, 2021.

[20] K. Lee, I. Fischer, A. Liu, Y. Guo, H. Lee, J. Canny, and S. Guadarrama, "Predictive Information Accelerates Learning in RL," in *Proc. NeurIPS*, 2020.

[21] K. Rakelly, A. Gupta, C. Florensa, and S. Levine, "Which Mutual-Information Representation Learning Objectives are Sufficient for Control?" in *Proc. NeurIPS*, 2021.

[22] S. Li, L. Zheng, J. Wang, and C. Zhang, "Learning Subgoal Representations with Slow Dynamics," in *Proc. ICLR*, 2021.

[23] S. Li, J. Zhang, J. Wang, Y. Yu, and C. Zhang, "Active Hierarchical Exploration with Stable Subgoal Representation Learning," in *Proc. ICLR*, 2022.

[24] V. Pong, M. Dalal, S. Lin, A. Nair, S. Bahl, and S. Levine, "Skew-Fit: State-Covering Self-Supervised Reinforcement Learning," in *Proc. ICML*, 2020, pp. 7783–7792.

[25] Z. Qian, M. You, H. Zhou, and B. He, "Weakly Supervised Disentangled Representation for Goal-Conditioned Reinforcement Learning," *IEEE Robotics Autom. Lett.*, vol. 7, no. 2, pp. 2202–2209, 2022.

[26] R. Islam, H. Zang, A. Goyal, A. Lamb, K. Kawaguchi, X. Li, R. Laroche, Y. Bengio, and R. T. des Combes, "Discrete Compositional Representations as an Abstraction for Goal Conditioned Reinforcement Learning," in *NeurIPS*, 2022.

[27] A. Goyal, R. Islam, D. Strouse, Z. Ahmed, H. Larochelle, M. Botvinick, Y. Bengio, and S. Levine, "InfoBot: Transfer and Exploration via the Information Bottleneck," in *Proc. ICLR*, 2019.

[28] D. Shah, B. Eysenbach, N. Rhinehart, and S. Levine, "Rapid Exploration for Open-World Navigation with Latent Goal Models," in *Proc. CoRL*, vol. 164, 2021, pp. 674–684.

[29] N. Tishby, F. C. Pereira, and W. Bialek, "The Information Bottleneck Method," in *Allerton Conference on Communication, Control, and Computing*, 1999.

[30] T. Zhang, S. Guo, T. Tan, X. Hu, and F. Chen, "Generating Adjacency-Constrained Subgoals in Hierarchical Reinforcement Learning," in *Proc. NeurIPS*, 2020.

[31] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[32] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *Proc. ICLR*, 2014.

[33] W. Bialek and N. Tishby, "Predictive Information," *arXiv preprint cond-mat/9902341*, 1999.

[34] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep Variational Information Bottleneck," in *Proc. ICLR*, 2017.

[35] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," *arXiv*, vol. abs/1807.03748, 2018.

[36] M. Federici, A. Dutta, P. Forré, N. Kushman, and Z. Akata, "Learning Robust Representations via Multi-View Information Bottleneck," in *Proc.ICLR*, 2020.

[37] N. J. Beaudry and R. Renner, "An Intuitive Proof of the Data Processing Inequality," *arXiv preprint arXiv:1107.0740*, 2011.

[38] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. R. Bradski, K. Konolige, and N. Navab, "Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes," in *Proc. ACCV*, vol. 7724, 2012, pp. 548–562.

[39] P. Ammirato, P. Poirson, E. Park, J. Kosecka, and A. C. Berg, "A Dataset for Developing and Benchmarking Active Vision," in *Proc. ICRA*, 2017, pp. 1378–1385.

[40] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proc. EMNLP-IJCNLP*, 2019, pp. 3980–3990.

[41] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, "OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework," *arXiv preprint arXiv:2202.03052*, 2022.

[42] P. Suraj, "Question Generation using Transformers," https://github.com/patil-suraj/question-generation, access on 2022-08-10.

[43] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-Level Control through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[44] S. Wold, K. Esbensen, and P. Geladi, "Principal Component Analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.