Demonstration-free Autonomous Reinforcement Learning via Implicit and Bidirectional Curriculum

Jigang Kim [*1 2] **Daesol Cho** [*1 2] **H. Jin Kim** [1 3]

Abstract

While reinforcement learning (RL) has achieved great success in acquiring complex skills solely from environmental interactions, it assumes that resets to the initial state are readily available at the end of each episode. Such an assumption hinders the autonomous learning of embodied agents due to the time-consuming and cumbersome workarounds for resetting in the physical world. Hence, there has been a growing interest in autonomous RL (ARL) methods that are capable of learning from non-episodic interactions. However, existing works on ARL are limited by their reliance on prior data and are unable to learn in environments where task-relevant interactions are sparse. In contrast, we propose a demonstration-free ARL algorithm via **I**mplicit and **B**i-directional **C**urriculum (**IBC**). With an auxiliary agent that is conditionally activated upon learning progress and a bidirectional goal curriculum based on optimal transport, our method outperforms previous methods, even the ones that leverage demonstrations.

1. Introduction

Reinforcement learning (RL) has enabled interactive agents to learn complex skills in various domains with little to no prior knowledge (Andrychowicz et al., 2020; Baker et al., 2019; Vinyals et al., 2019; Degrave et al., 2022). However, existing algorithms assume an episodic setting where each trial begins from a state sampled from some fixed initial state distribution, and they are not designed to learn autonomously in the real world which involves continual, uninterrupted interaction. The challenge of applying RL in the real world often arises in robotics where the practitioner has to bridge the gap between the tools available (episodic RL) and the non-episodic nature of real-world learning. In most cases, a multitude of time-consuming and costly external interventions such as human supervision, task-specific scripted policies, and custom experimental setups are deployed to reset the environment after each trial (Kumar et al., 2016; Ha et al., 2020; Nagabandi et al., 2020). These challenges should be addressed from the algorithmic level by developing RL agents that can learn autonomously with minimal interventions.

Previous works on RL agents in the real world primarily involve a mechanism to handle resets and may leverage prior data along with additional consideration for reward assignment. Reset mechanisms that prevent interventions by requesting a reset when necessary (Eysenbach et al., 2017; Kim et al., 2022) are only viable if manual resets are readily available. Under the non-episodic autonomous RL (ARL) framework (Sharma et al., 2021b), however, manual resets are not available on-demand and the agent must learn from continual interactions with no interventions. To overcome the challenge of the non-episodic setting, many previous methods rely on some form of prior data with varying degrees of privilege, ranging from the expert or sub-optimal trajectories (Sharma et al., 2022; Chen et al., 2022) to examples of states of interest (Zhu et al., 2020). However, a *truly* autonomous agent should be able to learn from scratch without external interventions and prior data. To that end, we propose an ARL algorithm that can train a goal-conditioned RL policy without demonstrations under the non-episodic, sparse reward setting.

It has been well established that existing RL algorithms do not perform well in the non-episodic setting (Co-Reyes et al., 2020) since the agent is unable to repeatedly practice for the evaluation task. A common framework for extending conventional RL to the non-episodic setting is to alternate between multiple objectives within one continual interaction, effectively dividing it into multiple episodes. Typically, the forward episode attempts the original objective and the backward episode follows an auxiliary objective that provides an anchor for the forward episode with a good

[*]Equal contribution , order decided by a coin toss. [1]Seoul National University [2]Artificial Intelligence Institute of Seoul National University (AIIS) [3]Automation and Systems Research Institute (ASRI). Correspondence to: Jigang Kim <jgkim2020@snu.ac.kr>, Daesol Cho <dscho1234@snu.ac.kr>, H. Jin Kim <hjinkim@snu.ac.kr>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

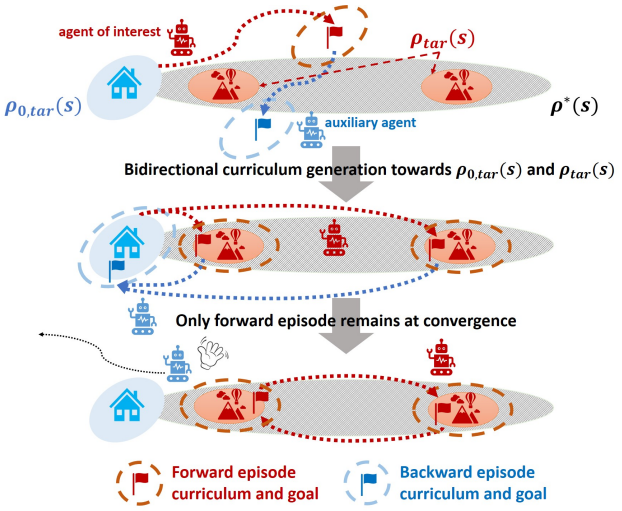


Figure 1. IBC proposes a bidirectional curriculum for both forward and backward episodes. The auxiliary agent is no longer activated after the agent of interest becomes capable.

initialization. An obvious choice for the auxiliary objective is to return to the initial state distribution (Eysenbach et al., 2017). However, this is not always the optimal choice as it wastes valuable transitions on returning all the way back to the initial state. Instead, it can be set to match other distributions such as the states observed in expert demonstrations, (Sharma et al., 2022) or maximize the state diversity (Zhu et al., 2020) for better sample efficiency or robustness.

We consider a conditionally activated auxiliary agent that returns to the initial state based on our observation that providing a strong anchor is crucial, especially if the task of interest involves an interaction that is sparse and unlikely to occur by chance in the non-episodic setting. Under the proposed method, the agent of interest is initially dependent on the auxiliary agent but becomes less reliant on it as training progresses in an implicit curriculum. When the agent of interest becomes capable, forward episodes can be rolled out consecutively without the auxiliary agent intervening and more transitions are devoted to training the agent of interest leading to better sample efficiency. While the auxiliary agent initially provides a strong foundation, additional guidance is needed to successfully train the agent of interest. Since the agent of interest is goal-conditioned and must learn without prior data, we generate curriculum goals that do not rely on demonstrations or predetermined curriculum. Specifically, we propose a bidirectional goal curriculum scheme to simultaneously select appropriate goals for the forward (agent of interest) and backward (auxiliary agent) episodes. To do so, we employ a curriculum based on the optimal transport between the desired goals and the candidate states sampled from past trajectories in the replay buffer to jointly optimize over the forward and backward curriculum goals.

The main contribution of our work is in proposing a demonstration-free ARL algorithm via **I**mplicit and **B**i-directional **C**urriculum (**IBC**). Evaluations in established ARL benchmarks and in RL environments modified for the ARL setting show that our method outperforms existing methods. Further analyses and ablation studies reveal that the proposed implicit curriculum (auxiliary agent) and explicit curriculum (bidirectional goal curriculum) are well-formed and necessary to successfully learn in the demonstration-free, non-episodic setting. To summarize, the key takeaways from our work are as follows:

- To the best of our knowledge, IBC is the first algorithm for non-episodic RL that can consistently learn without manual resets and demonstrations by leveraging curriculum learning.
- We propose a conditionally activated auxiliary agent and a bidirectional goal curriculum based on optimal transport to guide the agent of interest.
- In various environments, IBC achieves state-of-the-art performance against previous methods, including even the ones that leverage prior data.

2. Related Works

Autonomy in RL has gained much interest as RL is increasingly applied to various real-world robotics applications. Many practical applications adopted task-specific workarounds to implement resets in the real world with varying levels of automation – from human supervision and custom experiment setups (Yahya et al., 2017; Zeng et al., 2020) to scripted actions and pre-trained networks (Sharma et al., 2020; Thananjeyan et al., 2021). Since then, several algorithms for reset-free RL have been proposed, drawing inspiration from various topics in RL such as multi-task RL (Gupta et al., 2021; Walke et al., 2021), multi-stage RL (Smith et al., 2020; Xu et al., 2022), curriculum learning (Sharma et al., 2021a), and unsupervised skill discovery (Xu et al., 2020). Recently, the autonomous RL (ARL) framework formally defined the non-episodic RL setting. Several works have sought to address some variation of the ARL problem such as leveraging prior data or human preference to enable single-life and lifelong learning (Chen et al., 2022; Lu et al., 2020) or to handle irreversibility in the environment (Xie et al., 2022). However, we notice a lack of *truly* autonomous agents in existing methods and propose an ARL algorithm for demonstration-free, non-episodic RL in ergodic environments.

A common framework for replacing manual resets is to alternate between forward and backward episodes (Han et al., 2015). Not all such methods are capable of non-episodic RL

as some of them are not entirely reset-free and instead focus on reducing manual resets through backward episodes (Eysenbach et al., 2017; Kim et al., 2022). Reset-free methods that learn a separate policy to reset to diverse initial states (Zhu et al., 2020; Xu et al., 2020) are only viable in environments where task-relevant interactions are likely to occur by chance and either require prior data such as states of interest or are geared towards acquiring behavior primitives for downstream tasks. MEDAL (Sharma et al., 2022) and VaPRL (Sharma et al., 2021a) are directly comparable to our method, but MEDAL returns to the state distribution of the optimal policy instead of the initial state which requires expert demonstrations. While VaPRL can technically operate without demonstrations, we have found that removing them results in significant performance degradation in practice. This is likely due to the subgoal curricula scheme proposed in VaPRL which relies on demonstrations both for gathering good goal candidates and in calculating the cost for the goal selection process.

Curriculum learning has been deployed in RL to improve sample efficiency, encourage exploration, and solve complex multi-stage tasks (Narvekar et al., 2020). Such strengths are also desirable in the non-episodic setting. Curriculum in episodic RL often involves distribution matching to some desired task distribution (Ren et al., 2019; Klink et al., 2022; Huang et al., 2022; Cho et al., 2023) and task difficulty (Florensa et al., 2018; Sukhbaatar et al., 2017; Portelas et al., 2020; Jiang et al., 2021). However, these methods are not designed for the non-episodic setting. To address this, we propose an auxiliary agent and bidirectional goal curriculum to incorporate both task difficulty and task distribution matching. The auxiliary agent gradually fades away in an implicit curriculum conditioned on the learning progress (success rate) of the agent of interest. To apply goal curriculum in the non-episodic setting, we generate curriculum goals not only for the task goal (forward episode) but also for the initial state (backward episode) based on the Wasserstein distance metric.

3. Preliminary

3.1. Autonomous Reinforcement Learning

We assume an ergodic environment for the demonstration-free, non-episodic setting, similar to many previous works on autonomous RL (ARL). We consider the Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{G}, \mathcal{A}, \mathcal{P}, r, \gamma, \rho_0)$, where $\mathcal{S}$ denotes the state space, $\mathcal{G}$ the goal space, $\mathcal{A}$ the action space, $\mathcal{P}(s'|s,a)$ the transition dynamics, γ the discount factor, and ρ_0 the initial state distribution of the evaluation setting. The learning algorithm $\mathbb{A}$ is defined as $\mathbb{A} : \{s_j, a_j, r_j, s_{j+1}\}_{j=0}^{t} \mapsto \{a_t, \pi_t(\cdot|s)\}$, which maps the collected data until time t to an action a_t to be applied during the non-episodic training and its current best guess of the optimal evaluation policy $\pi_t(\cdot|s)$.

Typical implementations of RL algorithms (episodic) involve thousands or millions of sampling $s_0 \sim \rho_0(s)$, which require manual resets at the end of every episode. However, under the ARL framework (non-episodic), the initial state $s_0 \sim \rho_0(s)$ is sampled only once at the beginning and the agent interacts with the environment through the actions a_t determined by the algorithm $\mathbb{A}$ until $t \to \infty$.

ARL defines the *Deployed Policy Evaluation metric*, which measures how fast the policy π_t improves in terms of the evaluation performance for a given task:

$$\mathbb{D}(\mathbb{A}) = \sum_{t=0}^{\infty} \Big[J(\pi^*) - J(\pi_t) \Big] \tag{1}$$

where $J(\pi) = \mathbb{E}_{\rho_0,\pi,\mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$, and π^* is the optimal policy. The goal of algorithm $\mathbb{A}$ is to minimize $\mathbb{D}(\mathbb{A})$ by learning as fast as possible.

3.2. Surrogate Objective for Curriculum-based RL

We replace the original RL objective with a surrogate objective to be utilized for curriculum generation in Section 4 and describe it in detail. Let $\mathcal{T}$ be the joint distribution of some initial state s_0 and goal g. Then, the original objective $\max_\pi J(\pi)$ can be represented as,

$$\max_{\pi} V^{\pi}(\mathcal{T}) := \mathop{\mathbb{E}}_{(s_0,g)\sim\mathcal{T}} \left[V^{\pi}(s_0, g)\right] \tag{2}$$

where $V^{\pi}(s_0, g)$ is the goal-conditioned value function.

Our approach relies on the following generalizability condition (Florensa et al., 2018; Luo et al., 2018; Asadi et al., 2018; Ren et al., 2019) that is characterized by the Lipschitz continuity-based assumption:

$$|V^{\pi}(\mathcal{T}') - V^{\pi}(\mathcal{T})| \leq L \cdot D(\mathcal{T}, \mathcal{T}') \tag{3}$$

where L is the Lipschitz constant and $D(\mathcal{T}, \mathcal{T}') = \inf_{\mu\in\Gamma(\mathcal{T},\mathcal{T}')}(\mathbb{E}_{\mu}[d((s_0, g), (s_0', g'))])$ is the Wasserstein distance based on the distance metric $d(\cdot,\cdot)$. $\Gamma(\mathcal{T}, \mathcal{T}')$ denotes the set of all possible transport plans μ.

Under Eq (3), optimizing Eq (2) can be relaxed into the following lower-bound maximization,

$$\max_{\mathcal{T},\pi} \Big[V^{\pi}(\mathcal{T}) - L \cdot D(\mathcal{T}, \mathcal{T}^*) \Big] \tag{4}$$

where $(s_0^*, g^*) \sim \mathcal{T}^*$ is the joint distribution of the target initial state s_0^* and target goal state g^*. Intuitively, it maximizes the policy performance and closeness to $\mathcal{T}^*$, which results in a task curriculum with increasing difficulty.

4. Method

For a *truly* autonomous RL without external interventions and human supervision, we introduce 1) a conditionally

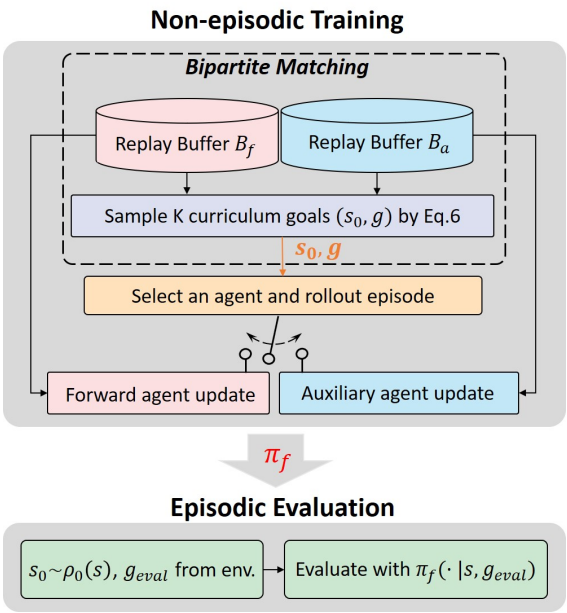


Figure 2. Overview of the proposed method, IBC.

activated auxiliary agent (π_a) that aids the forward agent (π_f) and 2) a bidirectional curriculum generation process for both forward and auxiliary agents that enables non-episodic RL without demonstrations via calibrated guidance.

4.1. Non-Episodic RL with an Auxiliary Agent

During non-episodic training, we alternate between the two agents such that the auxiliary agent guides the forward agent only when necessary. Specifically, we conditionally activate the auxiliary agent when the forward agent has failed at the given goal state such that the auxiliary agent gradually disappears as the forward agent improves which results in better sample efficiency. Let us consider the hypothetical setting where the forward agent is fully capable and the auxiliary agent does not intervene at all. Under this setting, the forward agent repeatedly attempts its target goal states $s_{g^*} \sim \rho_{tar}(s)$ without resets. Thus, the agent is no longer restricted by $\rho_0(s)$ unlike in episodic settings and we can consider a better initial state distribution by appropriately designing $\rho_{tar}(s)$.

Interestingly, a previous work (Kakade & Langford, 2002) provides theoretical grounds that $\rho_0(s)$ close to $\rho^*(s)$ enables efficient training in RL, where $\rho^*(s)$ denotes the state marginal distribution of the optimal policy π^*. If we set $\rho_{tar}(s)$ to be a subset of $\rho^*(s)$ from the optimal policy that achieves the evaluation goal g_{eval}, we can approximately satisfy this ideal initial state distribution. Note that the target goal s_{g^*} achieved by the forward agent policy π_f from the previous rollout becomes the initial state for the next rollout.

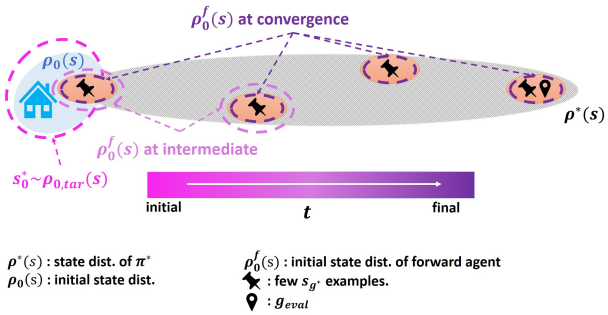


Figure 3. Visualization of $\rho_0^f(s)$ at various timesteps. As training progresses, the initial state distribution of the forward agent $\rho_0^f(s)$ gradually shifts from $\rho_0(s)$ to $\rho_{tar}(s)$.

In practice, it suffices for $\rho_{tar}(s)$, which is only used for bidirectional curriculum and not for RL, to contain a minimal number of key points that roughly outline the task to be adequate for the goal curriculum generation. This is because the curriculum goals effectively "fill in the blanks" by proposing past states from the replay buffer that are close to $\rho_{tar}(s)$. Typically, specifying $\rho_{tar}(s)$ requires only a handful of samples (~ 10) from $\rho_0(s)$ and g_{eval} combined to approximate $\rho^*(s)$. For some tasks, it suffices to specify $\rho_{tar}(s)$ with a single example from $\rho_0(s)$ and g_{eval} each. Unlike previous ARL methods, we do not require demonstrations with thousands of transitions or access to the expert policy.

Until now, we have considered the setting where π_f has converged and is fully capable. However, most of the rollouts by π_f before convergence will lead the agent to an arbitrary state rather than s_{g^*}, leading to highly-varying initial states for the next rollout which results in unstable learning. For this reason, we need an auxiliary agent that provides an anchor and guides the forward agent. More precisely, the auxiliary agent tries to bring the forward agent back to the set of target initial states $s_0^* \sim \rho_{0,tar}(s)$. Even though $\rho_{0,tar}(s)$ can be an arbitrary set of states that are useful for the repeated practice of the forward agent, we set $\rho_{0,tar}(s)$ to include the environmental initial state distribution $\rho_0(s)$. This is because providing a strong anchor is crucial in practice and the evaluation will be performed from $\rho_0(s)$.

While the proposed auxiliary agent resembles the standard backward agent from previous literature, there are two key differences. First, the auxiliary agent is activated only when the forward agent fails at the given curriculum goal such that the initial state for the forward agent $\rho_0^f(s)$ gradually evolves from $\rho_0(s)$ to $\rho_{0,tar}(s)$ in an implicit curriculum as π_f improves (Figure 3). This is by design such that the forward agent approximately satisfies the theoretically-grounded ideal initial state condition at convergence. Second, the auxiliary agent does not directly return to $\rho_{0,tar}(s)$

which encompasses $\rho_0(s)$, but to the intermediate goal for s_0^* obtained from the bidirectional goal curriculum. Thus, the goals proposed to the auxiliary agent are more diverse than the typical backward agent.

Algorithm 1 IBC

Input: $\widehat{\mathcal{T}}$, $\widehat{\mathcal{T}}^*$, $\mathcal{B}_f$, π_f, Q^{π_f}, $\mathcal{B}_a$, π_a, Q^{π_a}
$s \sim \rho_0(s)$ by `env.reset()`, set $\mathcal{O}$ to $\{f\}$
$\mathcal{O}$ denotes agent selection ($\{f\}$:forward, $\{a\}$:auxiliary)
while not done **do**
  get curriculum goal $g_{\mathcal{O}} \sim \widehat{\mathcal{T}}$
  while until (reach $g_{\mathcal{O}}$ or max episode steps) **do**
    $a \leftarrow \pi_{\mathcal{O}}(\cdot|s, g_{\mathcal{O}})$
    $s' \leftarrow \mathcal{P}(s'|s,a), r \leftarrow r(s,a,g_{\mathcal{O}})$
    $\mathcal{B}_{\mathcal{O}} \leftarrow \mathcal{B}_{\mathcal{O}} \cup (s, g_{\mathcal{O}}, a, r, s')$
    update $\pi_{\mathcal{O}}, Q^{\pi_{\mathcal{O}}}$
    $s \leftarrow s'$
  end while
  for once every N iteration **do**
    update $\widehat{\mathcal{T}}$ according to Eq (4) by solving Eq (6)
  end for
  if $\mathcal{O}$ was $\{a\}$ **then** set $\mathcal{O}$ to $\{f\}$
  else if π_f succeeded **then** keep $\mathcal{O}$ as $\{f\}$
  else set $\mathcal{O}$ to $\{a\}$
end while

4.2. Bidirectional Curriculum Generation

While our non-episodic training process involving an auxiliary agent, $\rho_{0,tar}(s)$, and $\rho_{tar}(s)$ approximately satisfies the ideal initial state condition, it might not be sufficient for autonomous training in environments where target states are difficult to be achieved from scratch. Thus, we need to find intermediate goals that can guide the learning of the agent. To find such goals without relying on demonstrations, the candidates must be obtained from past trajectories with highly varying initial states due to non-episodic training. We propose a bidirectional goal curriculum based on the surrogate problem (Eq (4)) for both forward and auxiliary agents without relying on demonstrations in the non-episodic setting.

For autonomous curriculum generation, we sample the candidates for $\mathcal{T}$ from past states in the replay buffer $\mathcal{B}$. To prevent a degenerate solution in the curriculum selection process, a diversity constraint is incorporated such that for every trajectory $\tau = (s_0, ..., s_{t_{final}}) \in \mathcal{B}$, at most one state can be chosen for $\mathcal{T}$. Then, Eq (4) is transformed as follows,

$$\max_{\pi_f, \mathcal{T}} \left[V^{\pi_f}(\mathcal{T}) - L \cdot D(\mathcal{T}, \mathcal{T}^*) \right]$$
$$\text{s.t.} \sum_t \mathbb{1}\left[(s_0, \phi_f(s_t)) \in \mathcal{T}\right] \leq 1, \quad s_0, s_t \in \tau, \forall \tau \in \mathcal{B} \tag{5}$$

where $\phi(\cdot)$ is a mapping function that abstracts the state space into the goal space. To solve Eq (5), we iteratively update $\mathcal{T}$ and policies π_f, π_a until π_f achieves a desirable evaluation performance. The policy optimization is simply achieved by applying off-the-shelf RL algorithms such as SAC (Haarnoja et al., 2018). The optimization of $\mathcal{T}$ is defined by the Wasserstein Barycenter problem augmented with a value bias term.

Inspired by Ren et al. (2019), we enforce $\mathcal{T}$ and $\mathcal{T}^*$ to be a set of K particles ($|\widehat{\mathcal{T}}| = |\widehat{\mathcal{T}}^*| = K$) where $(s_0, g)^i \sim \widehat{\mathcal{T}}$, and $(s_0^*, \phi(s_{g^*}))^i \sim \widehat{\mathcal{T}}^*$, rather than parameterizing their distribution. Then, to address the Wasserstein Barycenter problem (Eq (5)) in the combinatorial setting, we assign candidates for $\widehat{\mathcal{T}}$ to $\widehat{\mathcal{T}}^*$ via the following bipartite matching problem:

$$\min_{\tau^i = \{s_t^i, \forall t\} \in \mathcal{B}} \sum_{(s_0^*, s_{g^*})^i} w\left((s_0^*, s_{g^*})^i, \tau^i\right) \tag{6}$$

where $w(\cdot,\cdot)$ becomes

$$w\left((s_0^*, s_{g^*})^i, \tau^i\right) := c \left\| \phi_a(s_0^{*,i}) - \phi_a(s_0^i) \right\|_2 + \min_t \left(\left\| \phi_f(s_{g^*}^i) - \phi_f(s_t^i) \right\|_2 - \frac{1}{L} V^{\pi_f}(s_0^i, \phi_f(s_t^i)) \right), \tag{7}$$

when we define the distance metric $d((s,g),(s',g'))$ from Eq (3) as $c\left\|\phi_a(s) - \phi_a(s')\right\|_2 + \|g - g'\|_2$ (c is a hyperparameter). With the costs w defined according to Eq (7), we can construct a bipartite graph $\mathbf{G}(\{\mathbf{V}_a, \mathbf{V}_b\}, \mathbf{E})$. Let $\mathbf{V}_a$ be the set of nodes representing candidates for $\widehat{\mathcal{T}}$ and $\mathbf{V}_b$ be the set of nodes for $\widehat{\mathcal{T}}^*$. The weights of the edges are defined as $\mathbf{E}(v_a, v_b) = -w(v_a, v_b)$, where $v_a \in \mathbf{V}_a$ and $v_b \in \mathbf{V}_b$.

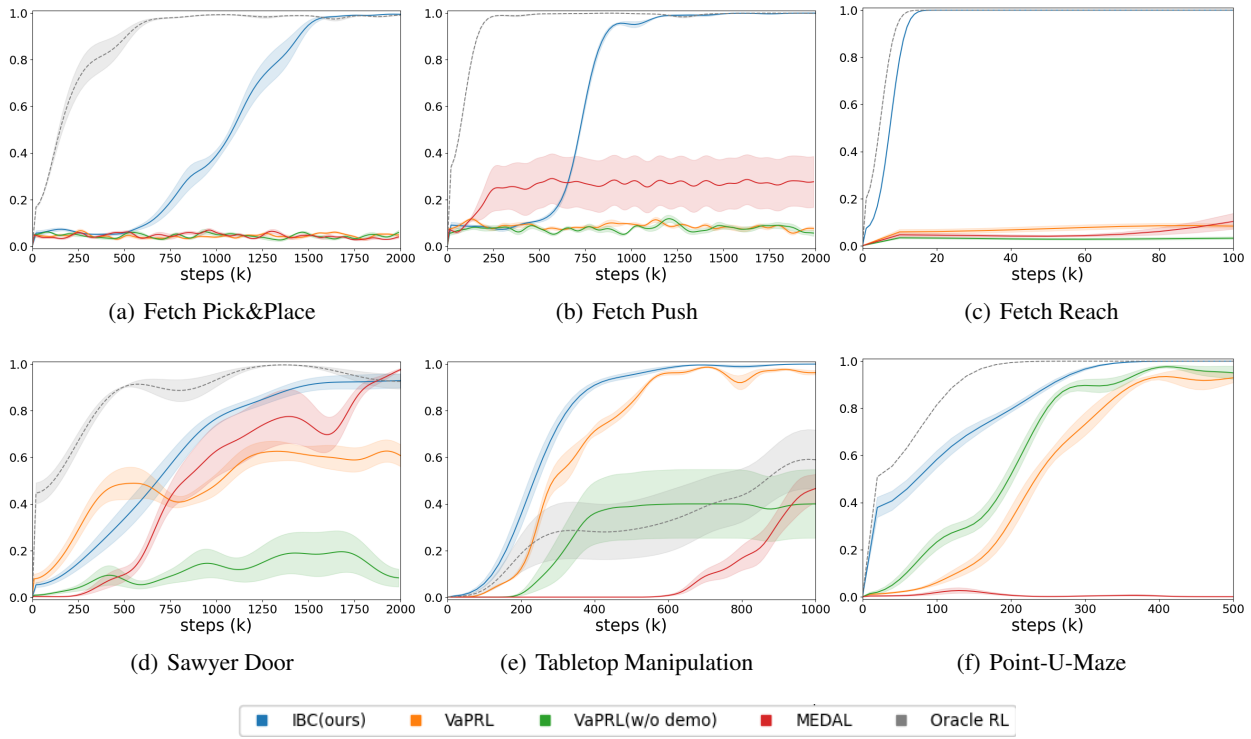
To solve the bipartite matching problem, the Minimum Cost Maximum Flow algorithm is utilized to find K edges with the minimum combined cost of connecting $\mathbf{V}_a$ and $\mathbf{V}_b$ (Ahuja et al., 1993). The resulting K forward curriculum goals will be proposed towards a region of the state space considered to be close to $s_{g^*} \sim \rho_{tar}(s)$ and within the capability of the forward agent as indicated by the value bias term. Similarly, the K auxiliary curriculum goals will be proposed towards a region considered to be close to $s_0^* \sim \rho_{0,tar}(s)$.

5. Experiment

We include six sparse reward environments to evaluate our method. Two environments – Tabletop Manipulation, Sawyer Door – are from established ARL benchmark, EARL (Sharma et al., 2021b), and the remaining four environments

Table 1. Conceptual comparison between our work and baseline algorithms.

	Demo-free	Curriculum	Agent Configuration	Backward Towards
oracle RL	✓	✗	single (SAC)	N/A
R3L	✓	✗	forward (VICE, Fu et al. (2018)) & backward (RND, Burda et al. (2018))	max$\mathcal{H}(s)$ for diverse states
VaPRL	✗	backward subgoal only (✓)	single (SAC)	$\rho_0(s)$
MEDAL	✗	✗	forward (SAC) & backward (VICE with $\rho^*(s)$)	$\rho^*(s)$ from expert demos
IBC(ours)	✓	both forward & backward (✓)	dual (forward & auxiliary) → single as training proceeds (SAC)	$\rho_{tar}(s)$ (a subset of $\rho^*(s)$)



(a) Fetch Pick&Place (b) Fetch Push (c) Fetch Reach

(d) Sawyer Door (e) Tabletop Manipulation (f) Point-U-Maze

Figure 4. Comparison of evaluation success rates of various algorithms. Shading indicates standard deviation across 5 seeds.

– Fetch environments (Plappert et al., 2018), Point-U-Maze – are modified versions of existing MuJoCo-based OpenAI Gym environments (Todorov et al., 2012; Brockman et al., 2016) for the ARL setting. These environments represent a mixture of robotic manipulation and locomotion tasks. Detailed descriptions of the environments are provided in Appendix A.

We compare with other previous methods designed for the ARL framework, which can be summarized as follows:

MEDAL (Sharma et al., 2022) – a backward agent that minimizes the distance between its state marginal distribution and the expert state distribution.

VaPRL (Sharma et al., 2021a) – value-based subgoal curricula towards the initial state distribution $\rho_0(s)$ during the backward episode; amenable to demonstration-free setting, but reports on the version with demonstration data.

oracle RL – a standard RL baseline such as SAC (Haarnoja et al., 2018) in an episodic setting with goal relabeling technique (Andrychowicz et al., 2017) common for sparse reward environments.

There exist other ARL methods such as R3L (Zhu et al., 2020) but we did not include them as they are already outperformed by VaPRL and MEDAL. We summarize the conceptual comparison between our method and previous ARL baselines in Table 1.

5.1. Results and Analyses

We follow the evaluation setting similar to the EARL benchmark (Sharma et al., 2021b). Specifically, the agent interacts with the environment after initially being spawned at $s_0 \sim \rho_0(s)$ and occasionally being reset to $s_0 \sim \rho_0(s)$ after hundreds of thousands of steps. Since we focus on minimizing the deployed policy evaluation metric, $\mathbb{D}(\mathbb{A})$, we report on $J(\pi_t)$ in 10k training step intervals by averaging returns from the policy over multiple evaluation

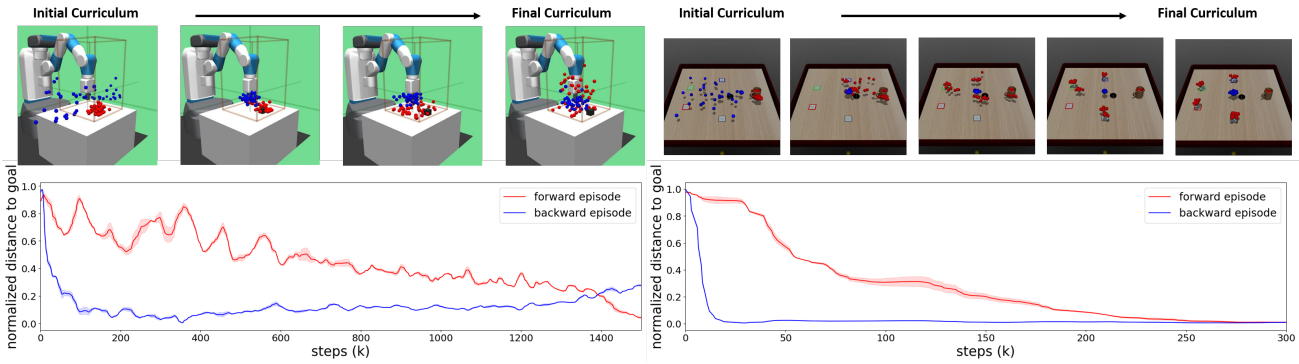


Figure 5. Visualization of the curriculum goals and their average normalized distance to assigned target goals (Left: Fetch Pick&Place, Right: Tabletop Manipulation). The red and blue dots indicate the curriculum goals for the forward and auxiliary agents, respectively. Note that the exact positions of the robots and objects are meaningless; these are just rendered from their default states.

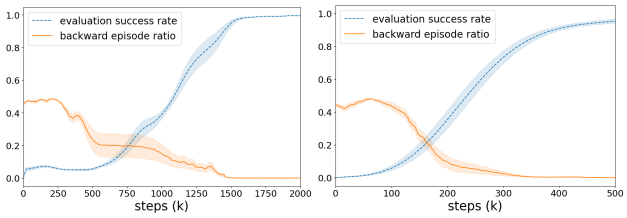
episodes. The code implementation of IBC and the instructions for reproducing the main result is available at https://github.com/snu-larr/ibc_official.

Evaluation results. As shown in Figure 4, the proposed method achieves state-of-the-art performance against other baselines, without requiring any demonstration data and even achieving comparable average return (success rate) to the oracle RL (episodic RL setting). Although some prior works such as VaPRL and MEDAL utilize nearly expert-level demonstration data, they have difficulty in environments where the task-relevant interactions are very sparse in the non-episodic setting or the evaluation goals g_{eval} are uniformly spread over some region rather than a few points such as Fetch environments. Furthermore, these methods are somewhat sensitive to the composition of the demonstration data in practice, which is detailed in Appendix B. For a fair comparison with our method, we also evaluated a version of VaPRL without demonstrations; it performed noticeably worse than the original VaPRL.

To validate whether the intervention of the auxiliary agent vanishes as training proceeds, we plot the episode ratio of the auxiliary agent within the latest 1k episodes. As shown in Figure 6, the auxiliary agent does not intervene when the forward agent is fully trained.

Bidirectional curriculum. To validate whether the bidirectional curriculum goals are properly interpolated and eventually converge to the desired target distributions, we evaluate the progress of the curriculum goals qualitatively and quantitatively. To do so, we visualize the forward and auxiliary curriculum goals and plot the corresponding normalized distance averaged over target goals assigned by bipartite matching (Section 4.2).

The plots in Figure 5 demonstrate that the average distance to goals consistently decreases as training proceeds, which



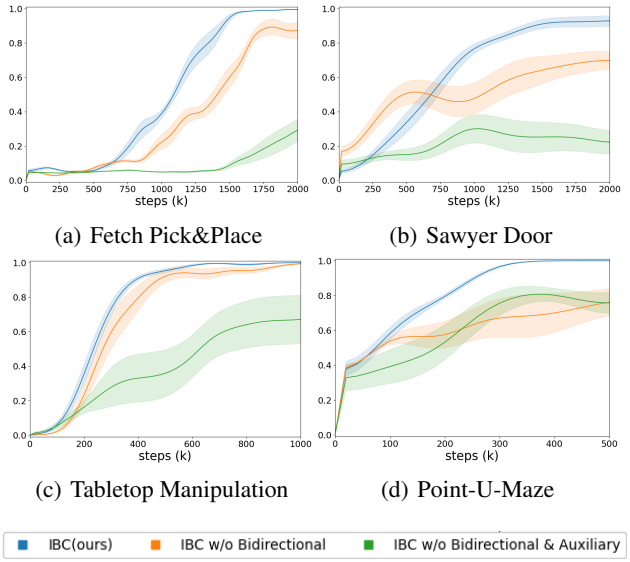
(a) Fetch Pick&Place (b) Tabletop Manipulation

Figure 6. Episode ratio of the auxiliary agent and evaluation success rate.

indicates that the curriculum goals for both forward and auxiliary agents have properly converged to their respective target states. The visualizations in Figure 5 provide further validation. Specifically, the forward curriculum goals gradually converge toward the $\rho_{tar}(s)$, which encompasses a region in the air and on the table for the Fetch Pick & Place, and five discrete points for the Tabletop Manipulation, respectively. The auxiliary curriculum goals also converge to the target goal states $\rho_{0,tar}(s)$, initially. However, there is a gradual shift of the auxiliary curriculum goals towards $\rho^*(s)$ after initial convergence which is reflected in the slight increase in average distance to goals for the backward episode ($\rho_{0,tar}(s)$), especially visible in the Fetch Pick & Place environment. This is because the candidates for the backward curriculum goals, which eventually become the initial states for the forward agent, are obtained from both $\rho_{0,tar}(s)$ and $\rho_{tar}(s) \subset \rho^*(s)$ when the forward agent remains at intermediate proficiency ($\sim$50%) for prolonged timesteps during training such as in Fetch Pick & Place, but less so in Tabletop Manipulation.

5.2. Ablation Study

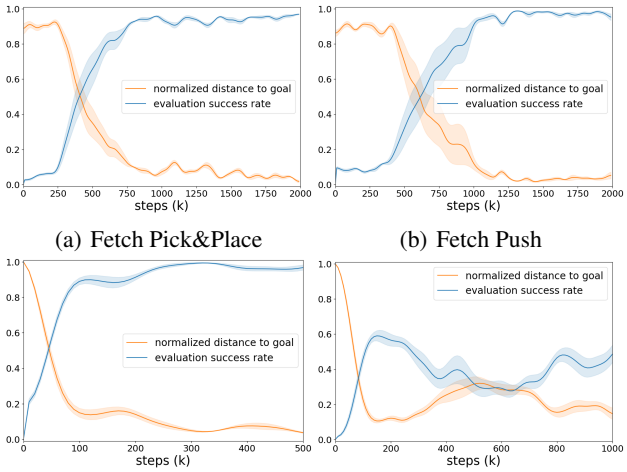
To investigate the role of the goal curriculum, we conduct an experiment without the proposed bidirectional curricu-



(a) Fetch Pick&Place (b) Sawyer Door

(c) Tabletop Manipulation (d) Point-U-Maze

Figure 7. Ablation study – removing the bidirectional curriculum and auxiliary agent proposed in this work degrades performance.



(a) Fetch Pick&Place (b) Fetch Push

(c) Point-U-Maze (d) Tabletop Manipulation

Figure 8. Normalized distance to goal and evaluation success rate for the reward-free variant.

lum (**IBC w/o Bidirectional**). We additionally ablate the auxiliary agent to validate its effectiveness (**IBC w/o Bidirectional & Auxiliary**). The latter corresponds to naive RL, where only a forward agent tries to optimize the task reward in a reset-free setting, without any backward episodes to return to some region such as $\rho_0(s)$.

As shown in Figure 7, there is consistent performance degradation in most of the environments for **IBC w/o Bidirectional**, which demonstrates the importance of gradually guiding the forward agent from easier initial states and goals to difficult ones. For **IBC w/o Bidirectional & Auxiliary**, there is an additional degradation in most of the environments, more so in object manipulation environments. The exact degree of the degradation may vary according to the given task and the choice of goal space mapping $\phi(s)$. In the case of Tabletop Manipulation, bidirectional curriculum generation does not have much effect on performance since the state space is relatively simple and does not benefit from intermediate curriculum goals. In the case of Point-U-Maze, the auxiliary agent is less effective since the initial state is quite far from the goal states.

5.3. Towards Reward-free Operation

To further enhance the autonomy of our method, we investigate the viability of the reward-free setting. While the proposed method operates on sparse rewards under the goal-conditioned setting which involves minimal human effort (defining the threshold for success), eliminating the need for explicit reward specification can be desirable, especially when dealing with high-dimensional inputs such as images. Prior work based on control as inference framework for future event matching (Fu et al., 2018; Eysenbach et al., 2021) enables reward-free methods that can also be applied to goal-conditioned RL such as C-learning (Eysenbach et al., 2020). We evaluate the performance of a variant of our method that replaces the SAC agent with a C-learning agent. For the sake of simplicity, we do not implement bidirectional curriculum goals for this variant but note that it is trivial to do so.

We additionally report the normalized distance to goal metric along with the success rate for this variant as it is based on future state matching and does not consider the threshold for success. Full results are available in Appendix B. In most environments, the C-learning variant achieves high success rates and even matches the proposed method in some environments. However, there are environments with low success rates, albeit with visible improvements in terms of the normalized distance to goal metric. This is because C-learning is reward-agnostic and tries to match the entirety of the desired future state, but some state elements may be more important than others for success depending on the task. Instead of matching in the state space, modifying the reward-free method to operate in a goal space that autonomously evolves towards the state elements that are most salient for the given task can be a promising approach for future research.

6. Conclusion

In this work, we considered a non-episodic RL setting where the agent should learn how to perform the given task autonomously without any external interventions such as man-

ual resets and prior data. We proposed IBC, a demonstration-free autonomous learning algorithm based on implicit and bidirectional curriculum generation. We have shown that our method outperforms previous methods, both in terms of sample efficiency and final average success rate. Our method is limited to reversible environments and still requires minimal human inputs for specifying sparse rewards. We'd like to build upon our method towards the reward-free setting, which has shown some promise in our results, by adopting the contextual MDP framework and devising a task-relevant goal space curriculum discovery for the reward-free setting.

7. Acknowledgement

This work was supported by the Korea Research Institute for Defense Technology Planning and Advancement (KRIT) Grant funded by the Defense Acquisition Program Administration(DAPA) (No. KRIT-CT-23-003, Development of AI tank crews based on deep reinforcement learning and establishment of virtual combat experiment)

References

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1st edition, 1993.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. Hindsight experience replay. *arXiv preprint arXiv:1707.01495*, 2017.

Andrychowicz, O. M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Asadi, K., Misra, D., and Littman, M. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 264–273. PMLR, 2018.

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., and Mordatch, I. Emergent tool use from multi-agent autocurricula. In *International Conference on Learning Representations*, 2019.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Chen, A. S., Sharma, A., Levine, S., and Finn, C. You only live once: Single-life reinforcement learning. *arXiv preprint arXiv:2210.08863*, 2022.

Cho, D., Lee, S., and Kim, H. J. Outcome-directed reinforcement learning by uncertainty & temporal distance-aware curriculum goal generation. *arXiv preprint arXiv:2301.11741*, 2023.

Co-Reyes, J. D., Sanjeev, S., Berseth, G., Gupta, A., and Levine, S. Ecological reinforcement learning. *arXiv preprint arXiv:2006.12478*, 2020.

Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de Las Casas, D., et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Eysenbach, B., Gu, S., Ibarz, J., and Levine, S. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. *arXiv preprint arXiv:1711.06782*, 2017.

Eysenbach, B., Salakhutdinov, R., and Levine, S. C-learning: Learning to achieve goals via recursive classification. In *International Conference on Learning Representations*, 2020.

Eysenbach, B., Levine, S., and Salakhutdinov, R. R. Replacing rewards with examples: Example-based policy search via recursive classification. *Advances in Neural Information Processing Systems*, 34:11541–11552, 2021.

Florensa, C., Held, D., Geng, X., and Abbeel, P. Automatic goal generation for reinforcement learning agents. In *International conference on machine learning*, pp. 1515–1528. PMLR, 2018.

Fu, J., Singh, A., Ghosh, D., Yang, L., and Levine, S. Variational inverse control with events: A general framework for data-driven reward definition. *Advances in neural information processing systems*, 31, 2018.

Gupta, A., Yu, J., Zhao, T. Z., Kumar, V., Rovinsky, A., Xu, K., Devlin, T., and Levine, S. Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6664–6671. IEEE, 2021.

Ha, S., Xu, P., Tan, Z., Levine, S., and Tan, J. Learning to walk in the real world with minimal human effort. *arXiv preprint arXiv:2002.08550*, 2020.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International*

conference on machine learning, pp. 1861–1870. PMLR, 2018.

Han, W., Levine, S., and Abbeel, P. Learning compound multi-step controllers under unknown dynamics. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6435–6442. IEEE, 2015.

Huang, P., Xu, M., Zhu, J., Shi, L., Fang, F., and Zhao, D. Curriculum reinforcement learning using optimal transport via gradual domain adaptation. *arXiv preprint arXiv:2210.10195*, 2022.

Jiang, M., Grefenstette, E., and Rocktäschel, T. Prioritized level replay. In *International Conference on Machine Learning*, pp. 4940–4950. PMLR, 2021.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer, 2002.

Kim, J., hyeon Park, J., Cho, D., and Kim, H. J. Automating reinforcement learning with example-based resets. *IEEE Robotics and Automation Letters*, 2022.

Klink, P., Yang, H., D'Eramo, C., Peters, J., and Pajarinen, J. Curriculum reinforcement learning via constrained optimal transport. In *International Conference on Machine Learning*, pp. 11341–11358. PMLR, 2022.

Kumar, V., Todorov, E., and Levine, S. Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 378–383. IEEE, 2016.

Lu, K., Grover, A., Abbeel, P., and Mordatch, I. Reset-free lifelong learning with skill-space planning. *arXiv preprint arXiv:2012.03548*, 2020.

Luo, Y., Xu, H., Li, Y., Tian, Y., Darrell, T., and Ma, T. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. *arXiv preprint arXiv:1807.03858*, 2018.

Nagabandi, A., Konolige, K., Levine, S., and Kumar, V. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pp. 1101–1112. PMLR, 2020.

Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., and Stone, P. Curriculum learning for reinforcement learning domains: A framework and survey. *arXiv preprint arXiv:2003.04960*, 2020.

Plappert, M., Andrychowicz, M., Ray, A., McGrew, B., Baker, B., Powell, G., Schneider, J., Tobin, J., Chociej, M., Welinder, P., et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

Portelas, R., Colas, C., Hofmann, K., and Oudeyer, P.-Y. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In *Conference on Robot Learning*, pp. 835–853. PMLR, 2020.

Ren, Z., Dong, K., Zhou, Y., Liu, Q., and Peng, J. Exploration via hindsight goal generation. *Advances in Neural Information Processing Systems*, 32, 2019.

Sharma, A., Ahn, M., Levine, S., Kumar, V., Hausman, K., and Gu, S. Emergent real-world robotic skills via unsupervised off-policy reinforcement learning. *arXiv preprint arXiv:2004.12974*, 2020.

Sharma, A., Gupta, A., Levine, S., Hausman, K., and Finn, C. Autonomous reinforcement learning via subgoal curricula. *Advances in Neural Information Processing Systems*, 34:18474–18486, 2021a.

Sharma, A., Xu, K., Sardana, N., Gupta, A., Hausman, K., Levine, S., and Finn, C. Autonomous reinforcement learning: Formalism and benchmarking. *arXiv preprint arXiv:2112.09605*, 2021b.

Sharma, A., Ahmad, R., and Finn, C. A state-distribution matching approach to non-episodic reinforcement learning. *arXiv preprint arXiv:2205.05212*, 2022.

Smith, L., Dhawan, N., Zhang, M., Abbeel, P., and Levine, S. Avid: learning multi-stage tasks via pixel-level translation of human videos. In *Robotics: Science and Systems XVI, Virtual Event / Corvalis, Oregon, USA, July 12-16, 2020*, 2020.

Sukhbaatar, S., Lin, Z., Kostrikov, I., Synnaeve, G., Szlam, A., and Fergus, R. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*, 2017.

Thananjeyan, B., Balakrishna, A., Nair, S., Luo, M., Srinivasan, K., Hwang, M., Gonzalez, J. E., Ibarz, J., Finn, C., and Goldberg, K. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3):4915–4922, 2021.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

Walke, H. R., Yang, J. H., Yu, A., Kumar, A., Orbik, J., Singh, A., and Levine, S. Don't start from scratch: Leveraging prior data to automate robotic reinforcement learning. In *6th Annual Conference on Robot Learning*, 2021.

Xie, A., Tajwar, F., Sharma, A., and Finn, C. When to ask for help: Proactive interventions in autonomous reinforcement learning. *arXiv preprint arXiv:2210.10765*, 2022.

Xu, K., Verma, S., Finn, C., and Levine, S. Continual learning of control primitives: Skill discovery via reset-games. *Advances in Neural Information Processing Systems*, 33: 4999–5010, 2020.

Xu, K., Hu, Z., Doshi, R., Rovinsky, A., Kumar, V., Gupta, A., and Levine, S. Dexterous manipulation from images: Autonomous real-world rl via substep guidance. *arXiv preprint arXiv:2212.09902*, 2022.

Yahya, A., Li, A., Kalakrishnan, M., Chebotar, Y., and Levine, S. Collective robot reinforcement learning with distributed asynchronous guided policy search. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 79–86. IEEE, 2017.

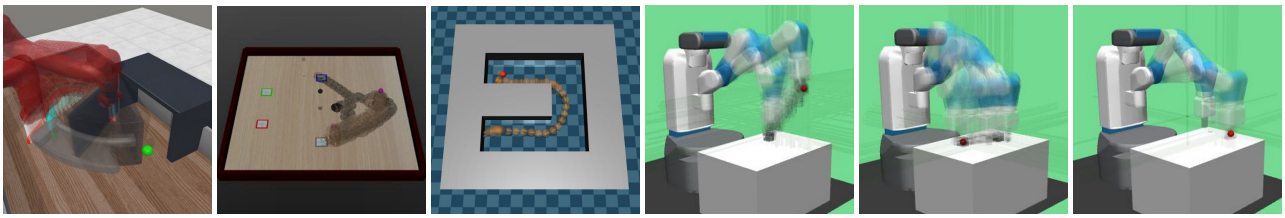
Zeng, A., Song, S., Lee, J., Rodriguez, A., and Funkhouser, T. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4): 1307–1319, 2020.

Zhu, H., Yu, J., Gupta, A., Shah, D., Hartikainen, K., Singh, A., Kumar, V., and Levine, S. The ingredients of real-world robotic reinforcement learning. *arXiv preprint arXiv:2004.12570*, 2020.

A. Experimental Details

A.1. Environment

- **Sawyer Door**: We use the original EARL benchmark environment (Sharma et al., 2021b). The evaluation goal state g_{eval} is the state where the door is fully closed and the target goal states $s_{g^*} \sim \rho_{tar}(s)$ during the non-episodic training are set to the corresponding states for door hinge angles between -60 degrees (open) and 0 degrees (closed).
- **Tabletop Manipulation**: We use the original EARL benchmark environment (Sharma et al., 2021b). The evaluation goal states consist of 4 discrete points and the target goal states are 5 discrete points (4 discrete evaluation goal points + 1 initial state of the object).
- **Point-U-Maze**: We used the 12 × 12 U-shaped maze environment where the initial position of the agent is $[0, 0]$ and the evaluation goal position is at the other end of the maze located at $[0, 8]$. The target goal states are randomly sampled in the feasible (not interfering with the maze walls, free space) state space.
- **Fetch Pick&Place, Fetch Push**: We modified the original Fetch environments from the gym-robotics package to convert it to a reversible (ergodic) setting by defining a constraint on the block position. The evaluation goals and target goal states are identical to the original Fetch Pick&Place, Fetch Push environments.
- **Fetch Reach**: We use the original Fetch environment from the gym-robotics package, where the evaluation goals and target goal states are obtained from uniformly sampled states in the area encompassing a region in the air and on the table.



(a) Sawyer Door (b) Tabletop Manip. (c) Point-U-Maze (d) Fetch Pick&Place (e) Fetch Push (f) Fetch Reach

Figure 9. Environments used in this work.

A.2. IBC Implementation

Table 2. Hyperparameters for IBC

critic hidden dimension	512	discount factor γ	0.99
critic hidden depth	3	curriculum buffer $\mathcal{B}_c$ capacity (# of trajectories)	1000
critic target τ	0.01	# of curriculum candidates, K (# of trajectories)	50
critic target update frequency	2	curriculum update frequency (once every N episode)	20
actor hidden dimension	512	learning rate	1e-4
actor hidden depth	3	RL optimizer	ADAM
actor update frequency	2	init temperature α_{init} of SAC	0.5
RL batch size	512	replay buffer $\mathcal{B}$ capacity (# of transitions)	1e6
c in curriculum update	3	Lipschitz constant L	5

We use a goal-relabeling technique (Andrychowicz et al., 2017) with SAC for sparse reward, goal-conditioned RL. There is a separate trajectory-level buffer $\mathcal{B}_c$ for the bidirectional goal curriculum. The values of various hyperparameters are detailed in Table 2. We set the goal space transformation of the auxiliary agent ϕ_a to abstract the proprioceptive states (e.g. gripper position in manipulation tasks and agent position in navigation or reaching tasks), and the goal space transformation of the forward agent ϕ_f to abstract the object-centric states when available (e.g. object position in manipulation tasks and agent position in navigation or reaching tasks).

A.3. Baseline Implementations

The baseline algorithms are trained as follows,

- **VaPRL** (Sharma et al., 2021a): There is no official code implementation, so we implemented it ourselves. We closely followed the details in the original paper and validated whether we have properly implemented the algorithm by obtaining statistically similar results when using demonstrations to the ones reported in (Sharma et al., 2021b).
- **MEDAL** (Sharma et al., 2022): We follow the default setting in the original implementation from `https://github.com/architsharma97/medal`.
- **naive RL**: We train a single agent to reach the given goal state until success or pre-determined, environment-specific maximum episode steps. After that, the target goal is resampled without resetting and the agent repeats the above process for hundreds of thousands of steps. We use SAC (Haarnoja et al., 2018) with the goal relabeling technique (Andrychowicz et al., 2017).
- **oracle RL**: Standard episodic RL is applied. Specifically, we use SAC (Haarnoja et al., 2018) with the goal relabeling technique (Andrychowicz et al., 2017).

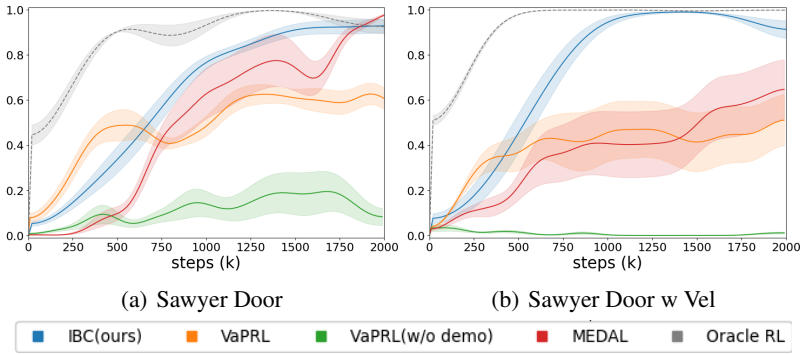
VaPRL and MEDAL require expert or near-expert demonstrations. For Sawyer Door and Tabletop Manipulation environments, we use the demonstration data (forward & backward episodes) provided by the EARL benchmark (Sharma et al., 2021b). For other environments, we collected demonstrations of similar quality (expert-level) and quantity (comparable amount of total timesteps) by rolling out the trained oracle RL policy.

B. Additional Experimental Results

B.1. Sawyer Door with Velocity Inputs

For the Sawyer Door environment in the EARL benchmark (Sharma et al., 2021b), we found instances of the door moving due to inertia even when the robot arm is not in contact. Without velocity information, this can violate the Markov Decision Process (MDP) assumption that the transition probability be fully observable. That is, a different next state s' can be obtained from the identical current state s and action a.

To alleviate it, we additionally experiment with the velocity-augmented state inputs. We concatenate the translational velocity of the door handle (3-dimensional) and train the agent with IBC and other baselines. The results in Figure 10 demonstrate that there are slight increases in the final average success rates and training stability with velocity-augmented states.



(a) Sawyer Door (b) Sawyer Door w Vel

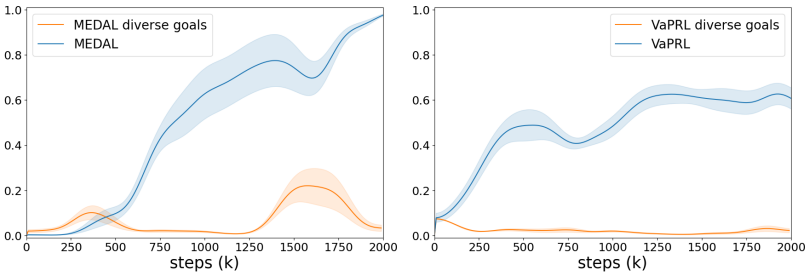
Figure 10. Experimental results for variants of Sawyer Door environments.

B.2. Sensitivity of Baseline Algorithms to the Demonstration Data

Although prior works (VaPRL, MEDAL) have shown some progress in developing a better ARL algorithm, these have some restrictions due to requiring demonstration data for selecting the curriculum subgoals or for computing the reward for the backward policy. Furthermore, we found that these baselines are somewhat sensitive to the composition of the demonstration data.

Specifically, we collected demonstration data for the Sawyer Door environment in two different ways. The first dataset consists of expert trajectories with fixed goal states (with trajectories of similar lengths), and the second dataset consists of expert trajectories with diverse goal states (with trajectories of varying lengths). Since we terminate the rollout right after the agent achieves the goal, trajectories may vary in length.

As shown in Figure 11, both VaPRL and MEDAL are somewhat sensitive to the composition of the data. It may be due to the subgoal selection strategy in VaPRL that is dependent on the length of the demonstration trajectory, or in the case of MEDAL, an imbalance in the expert state distribution (for training the backward reward) caused by trajectories of varying lengths due to the diverse goals.

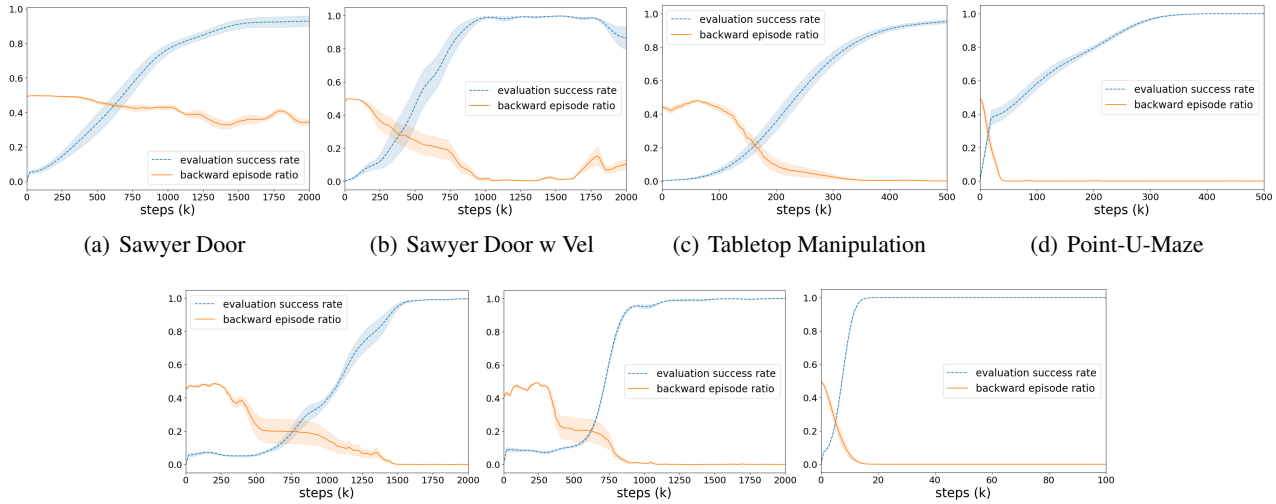


(a) MEDAL (b) VaPRL

Figure 11. Sensitivity of baselines to the composition of the demonstration data.

B.3. Episode Ratio of the Auxiliary Agent

We include the full results of the auxiliary agent episode ratio (Figure 12). The overall trend discussed in the main script also applies to the full result. We additionally report the results of Sawyer Door with velocity inputs as mentioned in B.1, and we have found that the backward episode ratio decreases when the velocity inputs are augmented.



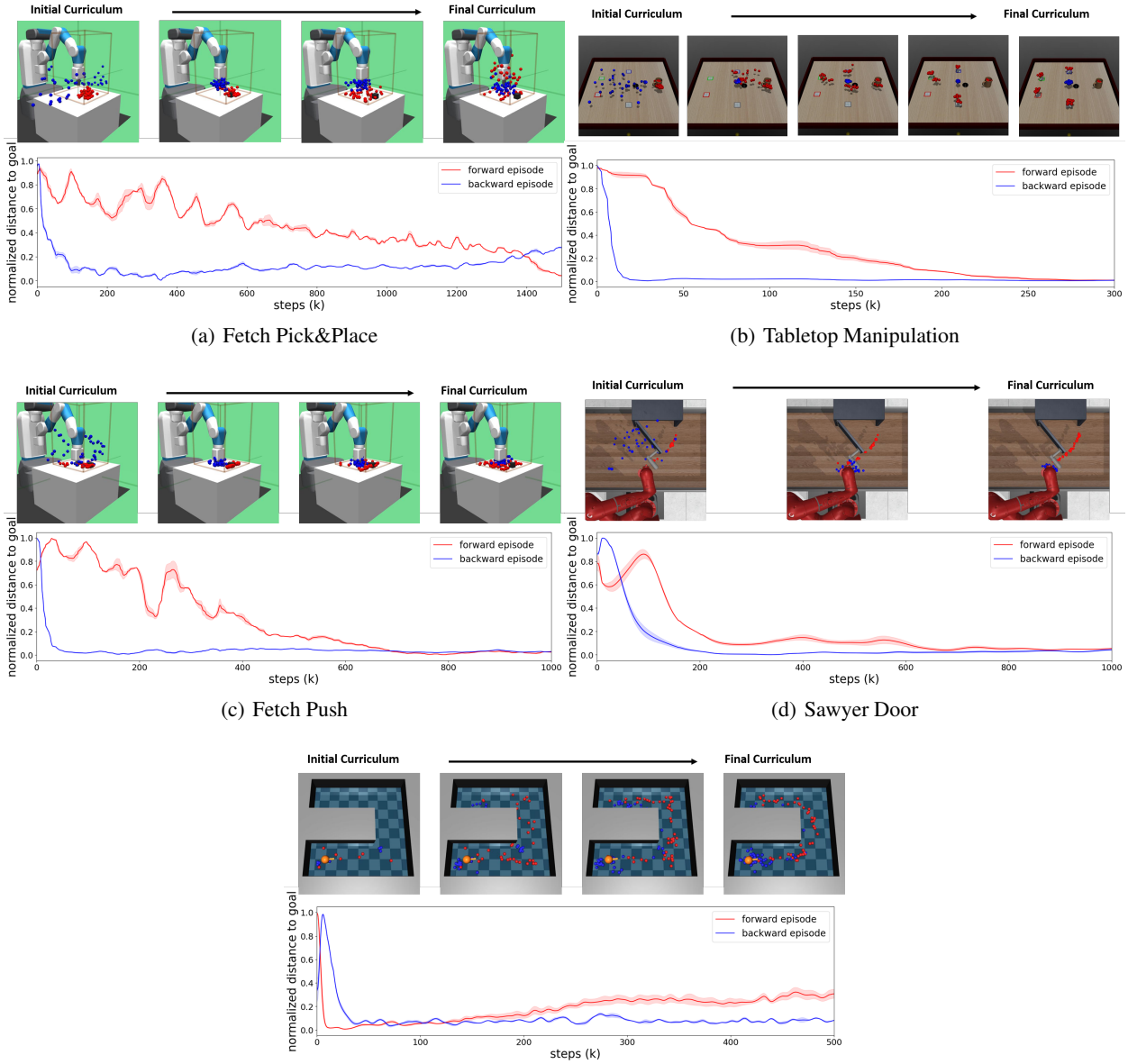
(a) Sawyer Door (b) Sawyer Door w Vel (c) Tabletop Manipulation (d) Point-U-Maze

(e) Fetch Pick&Place (f) Fetch Push (g) Fetch Reach

Figure 12. Full results of auxiliary agent episode ratio and evaluation success rate.

B.4. Curriculum Visualization

We report the full results for the curriculum visualization from the main script. As shown in Figure 13, curriculum goals for both forward and auxiliary agents converge to their respective target state distribution.



(a) Fetch Pick&Place

(b) Tabletop Manipulation

(c) Fetch Push

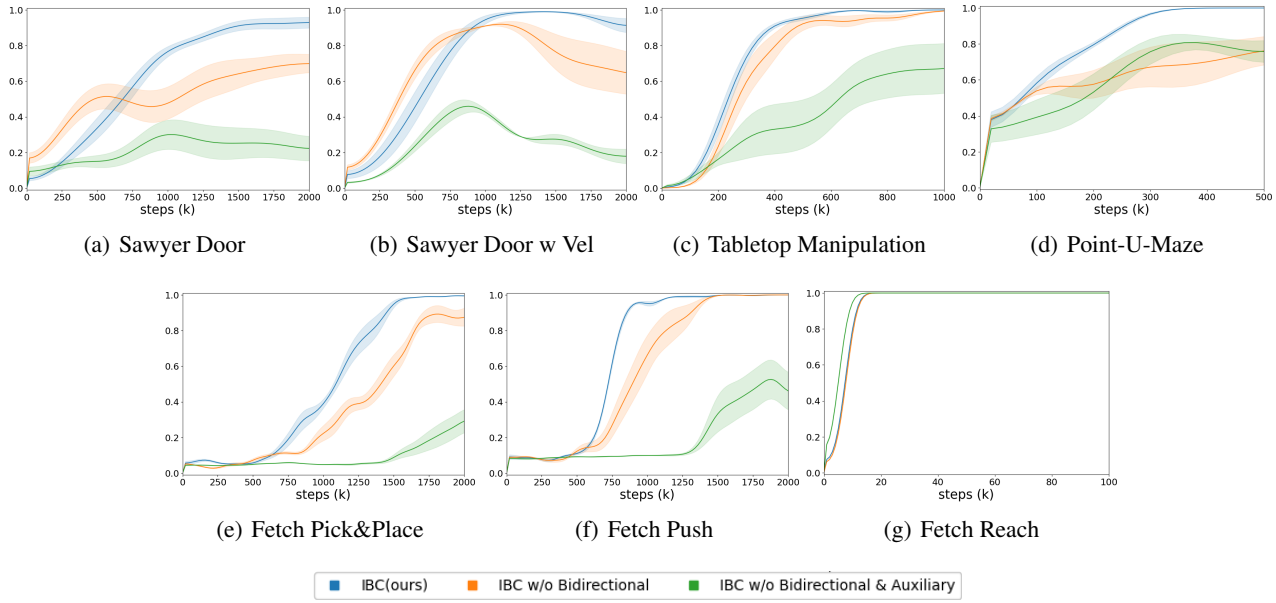
(d) Sawyer Door

(e) Point-U-Maze

Figure 13. Full results of the curriculum goals visualization and their average normalized distance to assigned target goals. The red and blue dots indicate the curriculum goals for the forward and auxiliary agents, respectively. Note that the exact positions of the robots and objects are meaningless; these are just rendered from their default states.

B.5. Ablation Study

We report the full results of the ablation study from the main script. As shown in Figure 14, success rates generally decrease as we remove the bidirectional goal curriculum, with further degradation when the auxiliary agent is removed.



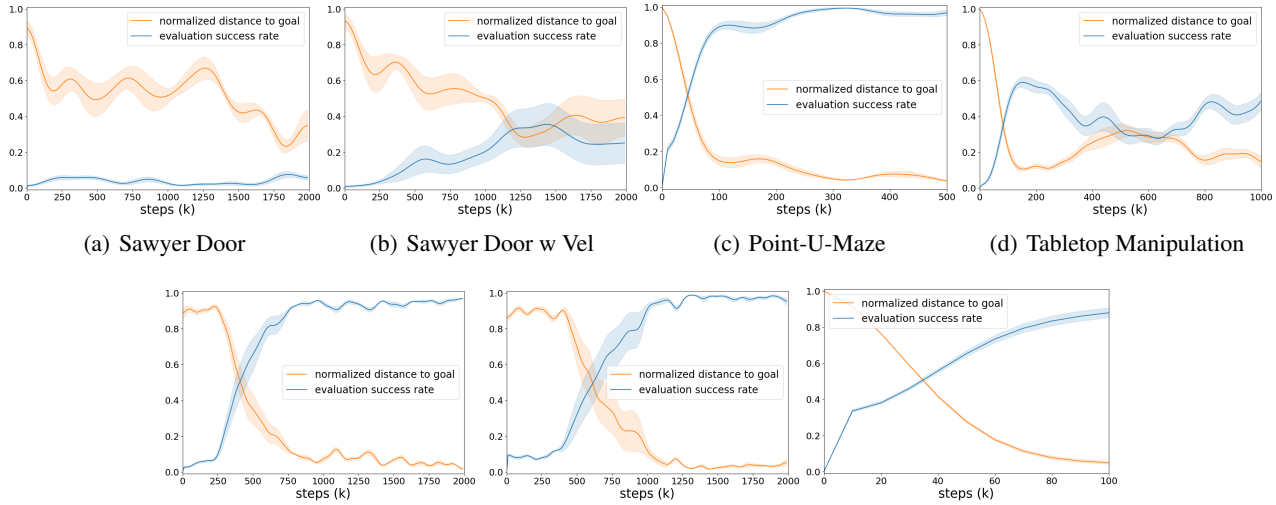
(a) Sawyer Door (b) Sawyer Door w Vel (c) Tabletop Manipulation (d) Point-U-Maze

(e) Fetch Pick&Place (f) Fetch Push (g) Fetch Reach

Figure 14. Full results of ablation study – removing the bidirectional curriculum and auxiliary agent proposed in this work degrades performance.

B.6. Reward-free Variant

We report the full results of the reward-free variant (C-learning variant) from the main script in Figure 15. The overall trend discussed in the main script also applies to the full result. One thing of note is that the reward-free variant performs better in the Sawyer Door environment when the door velocity is augmented.



(a) Sawyer Door (b) Sawyer Door w Vel (c) Point-U-Maze (d) Tabletop Manipulation

(e) Fetch Pick&Place (f) Fetch Push (g) Fetch Reach

Figure 15. Full results of the reward-free variant – normalized distance to goal and evaluation success rate.