TEACH: Temporal Variance-Driven Curriculum for Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

Reinforcement Learning (RL) has achieved significant success in solving single-goal tasks. However, uniform goal selection often results in sample inefficiency in multi-goal settings where agents must learn a universal goal-conditioned policy. Inspired by the adaptive and structured learning processes observed in biological systems, we propose a novel Student-Teacher learning paradigm with a Temporal Variance-Driven Curriculum to accelerate Goal-Conditioned RL. In this framework, the teacher module dynamically prioritizes goals with the highest temporal variance in the policy's confidence score, parameterized by the state-action value (Q) function. The teacher provides an adaptive and focused learning signal by targeting these high-uncertainty goals, fostering continual and efficient progress. We establish a theoretical connection between the temporal variance of Q-values and the evolution of the policy, providing insights into the method's underlying principles and convergence guarantees. Our approach is algorithm-agnostic and integrates seamlessly with existing RL frameworks. We demonstrate this through evaluation across 11 diverse robotic manipulation and maze navigation tasks. The results show consistent and significant improvements over state-of-the-art curriculum learning and goal-selection methods.

1 Introduction

Deep Reinforcement Learning (DRL) (Li, 2017) has been successfully applied in solving complex problems such as robotic manipulation (Han et al., 2023), flight control (Kaufmann et al., 2018), intelligent perception system (Chaudhary et al., 2023) and real-time strategy gameplay (Andersen et al., 2018). Building upon DRL's success, learning a generalized policy to solve multiple goal-oriented tasks is of huge interest. In the past, the multi-goal or multi-task (Tzannetos et al., 2023) problems have been addressed by merging Automatic Curriculum Learning (ACL) (Bengio et al., 2009; Karni et al., 1998; Portelas et al., 2020b; Akkaya et al., 2019; Team et al., 2021; Racaniere et al., 2019) with policy learning. Broadly, ACL resorts to presenting goal or task in increasing order of difficulty (Molina & Jouen, 1998; Yengera et al., 2021; Zhang et al., 2020), and has been successfully applied to Sim2Real transfer (Akkaya et al., 2019), continuously-parameterized environments (Portelas et al., 2020a), sequencing of components in multi-agent systems (Vinyals et al., 2019), and to improve DRL agent's sample efficiency and performance (Horgan et al., 2018; Flet-Berliac & Preux, 2019).

This paper focuses on ACL from the perspective of Goal Conditioned Reinforcement Learning (GCRL) (Schaul et al., 2015). Specifically, we address multi-goal sparse reward scenarios where the agent is trained to learn a universal policy to attain any goal in the goal space, and a binary reward is provided when the desired goal is achieved. In such multi-goal scenarios under ACL, learning is most effective when the agent engages with goals positioned at the skill frontier—neither too easy nor too difficult (Pinto et al., 2017; Sukhbaatar et al., 2017). Based on this observation, (Portelas et al., 2020a; Matiisen et al., 2019) use a curriculum that samples tasks with high Learning Progress (LP), while Value Disagreement Sampling (VDS) (Zhang et al., 2020) samples goals with high epistemic uncertainty. Although ACL has improved their performance, these approaches rely on noisy value estimates to design a curriculum, and these noisy estimates become inefficient in a high-dimensional goal space with sparse reward settings.

Concretely, we are motivated by theoretical analysis on ACL for linear regression models (Weinshall et al., 2018) and curriculum design for teachers via demonstration (Yengera et al., 2021). Our key insight is that relying on uniform goal sampling to update policy and value function neglects the intrinsic dynamics of learning, where certain goals require more focus due to their evolving contributions to policy improvement and value function learning. To address this, we propose a curriculum learning framework based on temporal variance in Q-values, highlighting regions where the learning dynamics are active. Although this curriculum is defined using Q-value estimates, we establish that it inherently captures the co-evolution of both the policy and value function, a connection we formalize mathematically to provide theoretical support.

Specifically, we introduce a Student-Teacher framework for GCRL that utilizes learning progress measured by temporal variance in policy confidence score to guide curriculum design. In this framework, the teacher module (goal proposer) evaluates the agent's policy and assigns a policy confidence score to each goal in the goal space. This policy confidence score, derived from state-action value (Q) estimates, corresponds to the expected return under the current policy for a given goal. The teacher leverages these confidence scores to dynamically design the curriculum. The teacher ensures that the curriculum continuously adapts to the agent's current capabilities and promotes efficient progress. To measure progress, we introduce the concept of Learning Progress (LP), which quantifies the temporal change in policy confidence scores. The teacher evaluates learning progress by monitoring the variance in these scores over time. This variance indicates the goals at the skill frontier, where the agent's policy is evolving most rapidly. This allows the teacher to focus on regions of the goal space where the agent's learning is most active, promoting continual and effective policy improvement.

The proposed algorithm has no static design heuristic Tzannetos et al. (2023); Eimer et al. (2021) and is ensemble-free (Zhang et al., 2020). We name our approach **TEACH** (**T**emporal varainc**E** driven **A**utomatic **C**urriculum teac**H**er). Further, to demonstrate the effectiveness of TEACH, we compare the proposed algorithm with Proximal Curriculum for Reinforcement Learning (ProCurl) (Tzannetos et al., 2023) and Self-Paced Context Evaluation (SPaCE) (Eimer et al., 2021) approaches, which use value estimates for curriculum learning in contextual multi-task scenarios. We adapt their strategies to a multi-goal setting. We also compare with current state-of-the-art VDS (Zhang et al., 2020) in multi-goal GCRL. Finally, We show that TEACH consistently improves performance on different challenging tasks, including robotic manipulation (Plappert et al., 2018), dexterous in-hand manipulation (Plappert et al., 2018), and Maze navigation (Zhang et al., 2020). We summarize our contributions as follows:

- Theoretical insights: We establish a formal theoretical connection between Q-values and policy evolution, demonstrating that changes in Q-values relate to policy divergence and help identify regions of significant policy evolution.
- Curriculum design: We propose a curriculum strategy based on Learning Progress (LP) that mitigates the impact of noisy value estimates, promoting efficient learning in multi-goal sparse reward scenarios.
- Convergence analysis: We provide a formal analysis of the convergence of our curriculum learning approach, ensuring stable and consistent policy improvement.
- Validation and robustness: We validate our method through extensive experiments across diverse robotic manipulation tasks, demonstrating its effectiveness and robustness across different settings.

2 Related Work

Curriculum learning has been recognized as a critical factor in addressing various machine learning challenges (Selfridge et al., 1985; Elman, 1993; Bengio et al., 2009; Cangelosi & Schlesinger, 2015), aiming to structure the presentation of samples during the learning process. Although the experts can manually create such curricula tailored to specific problems, the concept of Automatic Curricula Learning (Graves et al., 2017; Portelas et al., 2020a) concentrates on developing algorithms capable of autonomously arranging the sequences of learning problems to optimize agent performance.

Curriculum strategies for DRL: In DRL, applications of ACL are widespread as researchers often stumble to make generalist agents and search for strategies (Cobbe et al., 2020; Zhang et al., 2020; Rajeswaran et al., 2016) that can train agents beyond their initial success. Curriculum reinforcement learning techniques (Florensa et al., 2017; 2018; Portelas et al., 2020a; Klink et al., 2021; Zhang et al., 2020; Racaniere et al., 2019; Sukhbaatar et al., 2017; Portelas et al., 2020a; Jiang et al., 2021b; Li et al., 2023; Dennis et al., 2020; Jiang et al., 2021a) primarily concentrate on enhancing an agent's learning efficiency or effectiveness across a set of tasks. Curriculum Reinforcement Learning has also been successfully extended to sim-to-real transfer by adapting domain randomization (Akkaya et al., 2019). Exploiting world models for curriculum learning have also been explored by (Hu et al., 2023; Mendonca et al., 2021; Pong et al., 2019; Pitis et al., 2020). Planning Exploratory Goals (PEG) (Hu et al., 2023) is a goal-conditioned policy that optimizes directly for goals that would result in high exploratory value trajectories. However, our focus lies on ACL design for model-free GCRL.

Automatic curriculum learning for goal-conditioned RL is an approach for presenting goals to learning agents in a meaningful order (Schaul et al., 2015; Liu et al., 2022). Hindsight Experience Replay (HER) (Andrychowicz et al., 2017) is an implicit curriculum strategy that relabels unsuccessful trajectory rollout as successful. A more robust extension of it Curriculum-guided HER (CHER) (Fang et al., 2019) uses an adaptive relabelling strategy based on diversity and goal-proximity. Combined with HER, (Zhang et al., 2020) propose a Value Disagreement Sampling (VDS) that uses a goal proposer module that prioritizes goals that maximize the epistemic uncertainty of the value function. A learning-based strategy using a Generative Adversarial Network (GAN) was used by (Florensa et al., 2018), generating goals with intermediate success probability.

Learning progress-based curriculum strategies have also been studied, which sample goals more aggressively toward which agent shows the most progress. A student-teacher framework for discrete task space that samples tasks with high LP was presented by (Matiisen et al., 2019). To extend this to continuous task space (Portelas et al., 2020a) uses a Gaussian Mixture Model (GMM) on a tuple of tasks and absolute learning progress. The tasks are then sampled from a Gaussian chosen proportionally to its mean absolute learning progress. These methods use a dense reward structure and compute LP using either the nearest neighbour (Portelas et al., 2020a) or change in episodes return for a given task. This limits their direct applicability to GCRL with binary episodic returns with random initial states. SPaCE (Eimer et al., 2021) uses the values function to design a curriculum learning based on learning progress. However, their work targets contextual RL for discrete task settings. In a similar context, ProCurl (Tzannetos et al., 2023), inspired by the pedagogical concept of *Zone of Proximal Development* (Vygotsky & Cole, 1978), uses values estimates to design a curriculum strategy that sample task that is neither too easy nor too hard.

Existing curriculum RL methods often struggle with adaptively selecting goals that maximize learning progress, relying on static heuristics (Tzannetos et al., 2023; Eimer et al., 2021) or computationally expensive uncertainty (Zhang et al., 2020) measures derived from the value function. To address this, we propose a temporal variance-driven curriculum strategy that prioritizes goals based on the variability of the policy learning progress over recent time steps. By capturing the temporal divergence in the policy learning signal, our approach adaptively identifies goals at the edge of the agent's capability without relying on static heuristics or large ensembles, ensuring efficient and continual progress.

3 Formal Problem Setup

MDP Environment. We consider the learning environment defined as Markov Decision Process (MDP) $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma, H, \mathcal{R}, \mathcal{G})$. Where $\mathcal{S}, \mathcal{A}$ represents the state and action space of the process, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ represents the transition dynamics, H denotes the episode length, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ defines reward. The term $\mathcal{G} \subset \mathcal{S}$ specifies the pool of goals.

Multi-Goal RL Learning. We consider a problem where the RL agent can achieve multiple goals when acting via a stochastic policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ that maps a state to distribution over actions. Given a unique goal g sampled from goal space $\mathcal{G}$, the agent rolls out a trajectory to achieve success in the task via executing the stochastic policy π in the MDP $\mathcal{M}$. The agent is trained to maximize the expected discounted reward $\mathbb{E}_{s_{0:H-1}, a_{0:H-1}, r_{0:H-1}, g} \Sigma_{h=o}^{H-1} \gamma^h r_h$ obtained by policy π parametrized by θ. This multi-goal

Algorithm 1 Interplay of Teacher-Student components in RL agent training

1: **Initialization:** Initialize parameterized RL policy π_{θ_1}
2: **for** $t = 1, 2, ...$ **do**
3: Teacher (goal-proposer) picks a goal g from goal space $\mathcal{G}$.
4: Student (RL agent) generates a rollout trajectory τ_t aiming to reach the proposed goal g.
5: Update the parameterized RL agent π_{θ_t} to $\pi_{\theta_{t+1}}$.
6: **end for**

goal-conditioned problem can be framed as a standard RL problem with redefined state space $\mathcal{S} \times \mathcal{G}$.

Curriculum Learning. Further, we extended this multi-goal RL setup to a student-teacher paradigm. The agent's performance over goal g, which is uniformly sampled from goal space $\mathcal{G}$ can be defined as a policy confidence score $\mathcal{C}^\pi(g) = Q^\pi(s, g, a)$. In the context of our problem, we treat the policy as the student component and the goal proposer unit as the teacher component. The student component is updated using the following parameters: the current policy π_{θ_t}, the goal g sampled by the teacher from goal space $\mathcal{G}$, and the rollout trajectory $\tau_t = \{(s_t, a_t, \mathcal{R}(s_t, a_t)\}$. On the other hand, the the teacher component defines a curriculum over goal space to improve the student's performance. The focus of this work lies on the design of a teacher (curriculum) that can guide the student to learn in a sample efficient manner. The student-teacher interplay happens at the start of an episode, i.e., the teacher samples a target goal from the goal space for that particular episodic rollout. We summarize this student-teacher interplay in Algorithm 1.

4 Temporal Variance Driven Curriculum Design

This section introduces a novel curriculum learning approach that leverages policy confidence score to evaluate learning progress. Existing methods primarily rely on value estimate-based metrics (Zhang et al., 2020; Eimer et al., 2021; Tzannetos et al., 2023) to design curricula, which have shown promise in multi-task RL. However, value estimates are often noisy (Libardi et al., 2021; Raileanu & Fergus, 2021) because they are trained using small, random batches of data collected from rollouts generated by the agent, which is still learning. One potential way to mitigate this issue is to use Polyak averaging (Polyak & Juditsky, 1992; Damani & Pinto, 2023) to obtain smoother value functions (refer to experiment section). Nevertheless, we argue that the effect of noisy values becomes more pronounced when employing static heuristic-based strategies (Tzannetos et al., 2023; Eimer et al., 2021).

To address this, we propose leveraging the temporal evolution of the Q-function. We posit that even when value estimates are noisy, their temporal evolution can reveal a clear direction of improvement and suppress noise in curriculum learning. Further, the temporal variance in Q-values inherently captures the co-evolution of both the policy and value function, making it more informative and less susceptible to noisy value estimates. By focusing on this temporal signal, our strategy provides a more accurate representation of the agent's learning progress, ultimately enabling the generation of a more effective learning curriculum. To this end, we first establish the theoretical connection between Q-value and policy evolution. Next, we introduce the policy confidence score, a formal metric to quantify the performance of the policy, followed by curriculum design and convergence analysis.

4.1 Theoretical Connection between Q-value and Policy Evolution

The Q-function, $Q^\pi(s, a) = \mathbb{E}_{a_h \sim \pi_\theta(\cdot|s_h),\, s_{h+1} \sim \mathcal{T}(\cdot|s_h, a_h)} \left[\sum_{h=0}^{H-1} \gamma^h \mathcal{R}(s_h, a_h) \,\middle|\, s_0 = s, a_0 = a \right]$, represents the expected return from a given state-action pair under policy π_θ. Changes in the policy π_θ directly influence the Q-values through the Bellman operator (Bellman, 1966):

$$Q^\pi(s, a) = \mathcal{R}(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s, a)} \left[\mathbb{E}_{a' \sim \pi_\theta(\cdot|s')} Q^\pi(s', a') \right]. \tag{1}$$

As the policy evolves and improves, the state-action visitation distribution shifts, causing the Q-function to update and converge toward a new fixed point. In practical RL, Q-learning is often a proxy for policy iteration. Thus, any temporal variance in Q-values implicitly reflects changes in the policy, as policy improvement follows from Q-value updates. To understand this evolution, consider a soft policy update mechanism (Haarnoja et al., 2017) where:

$$\pi_{\theta_t}(a|s) \propto \exp\left(\frac{Q_t^{\pi}(s,a)}{\alpha}\right), \tag{2}$$

with $\alpha > 0$ acting as a temperature parameter controlling the degree of exploration. The Kullback-Leibler (KL) divergence (Kullback & Leibler, 1951) between consecutive policies can be expressed as:

$$\mathrm{KL}(\pi_{\theta_{t+1}} \parallel \pi_{\theta_t}) = \mathbb{E}_s \sum_a \pi_{\theta_{t+1}}(a|s) \log \frac{\pi_{\theta_{t+1}}(a|s)}{\pi_{\theta_t}(a|s)}. \tag{3}$$

Expanding this expression using the Q-values yields (refer to supplementary A for complete proof):

$$\mathrm{KL}(\pi_{\theta_{t+1}} \parallel \pi_{\theta_t}) \approx \frac{1}{2\alpha^2}\mathbb{E}_s\left[\mathrm{Var}_{a\sim\pi_{\theta_t}}(\Delta Q(s,a))\right]. \tag{4}$$

This result indicates that the KL divergence between successive policy updates is approximately proportional to the variance of Q-value changes under the current policy. Regions with high Q-value update variance correspond to significant policy shifts, highlighting the strong interplay between value estimation and policy adaptation. Moreover, tracking Q-value variance provides a natural signal for guiding policy updates, providing a dual perspective that captures both value and policy dynamics. This dual perspective provides a more robust signal by leveraging complementary information from value and policy updates. Exploiting this connection can lead to a more effective curriculum design strategy, guiding learning toward the most informative state-action regions.

In the next subsection, we mathematically define the policy confidence score, a formal metric that quantifies the expected future rewards of a policy and serves as an actionable measure of its performance, highlighting its critical role in guiding effective policy updates.

4.2 Policy Confidence Score

The RL policy aims to maximize the expected sum of future rewards over trajectories sampled from the current policy. The expected future reward of the current policy provides a natural measure of its performance. This intuition leads to the definition of a policy confidence score, which can be formalized as follows:

The objective of the learner agent is to maximize:

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\mathcal{R}(\tau)\right] \tag{5}$$

Where $\mathcal{R}(\tau)$ denotes the total reward from trajectory τ, and π_θ is the policy governing the agent's behavior. For each state-action pair $(s,a) \in S \times \mathcal{A}$ in the trajectory τ, the state-action values function $Q^\pi(s,a)$ is defined as:

$$Q^\pi(s,a) = \mathbb{E}_{a_h\sim\pi_\theta(\cdot|s_h),\, s_{h+1}\sim\mathcal{T}(\cdot|s_h,a_h)}\left[\sum_{h=0}^{H-1}\gamma^h \mathcal{R}(s_h,a_h)\,\middle|\, s_0 = s, a_0 = a\right] \tag{6}$$

Here, the expectation is computed with respect to $a_h \sim \pi_\theta(\cdot|s_h)$ and $s_{h+1} \sim \mathcal{T}(\cdot|s_h,a_h)$ for all $H > h \geq 0$. The goal of the learner is to find an optimal behavior strategy π_θ for the agent to obtain optimal rewards, which is defined as:

$$J(\theta) = \int_{s\in\mathcal{S}} \mathcal{T}^\pi(s) \int_{a\in\mathcal{A}} \pi_\theta(a|s) Q^\pi(s,a) \tag{7}$$

Algorithm 2 Temporal Variance Driven Curriculum Design

1: **Initialization:** Initialize parameterized RL policy π_θ, replay buffer R, goal space $\mathcal{G}$
2: Sample N goals from goal space $\mathcal{G}$
3: **for** $t = 1$ **to** K **do**
4: Compute $\mathcal{C}^\pi(g)$ using $[Q^\pi(s,a,g)]$ for all N goals
5: Compute $LP(g)$ using equation 10 for each goal
6: Sample target goal g and rollout goal-conditioned trajectory $\tau_t(\pi_\theta|s,g)$
7: Update policy parameter θ
8: **end for**

The gradient of the objective defined in Equation 7 with respect to θ is given by the policy gradient theorem (Sutton & Barto, 2018):

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[Q^\pi(s,a)\nabla_\theta \ln \pi_\theta(a|s)\right] \tag{8}$$

It is evident from equation 8 that the policy update is directly proportional to the expected future reward governed by $Q^\pi(s,a)$. Motivated by this and the theoretical insights provided above, which highlight the connection between Q-value and policy evolution, we define the policy confidence score ($\mathcal{C}^\pi(g) = Q^\pi(s,g,a)$), which measures future expected rewards and can track the performance of both policy and state-action value function.

4.3 Curriculum Design

Now, to design a curriculum strategy, we need to explicitly define a curriculum method $\mathcal{K}^\pi(g|s_0) \propto f(s_0,g)$, where f is a function that encodes curriculum objective. For example, in Value Disagreement Sampling (VDS) (Zhang et al., 2020), f is an epistemic uncertainty of goals in task space. Similarly, in SPaCE (Eimer et al., 2021), f is the learning progress defined as the difference in value function over time for a goal $g \in \mathcal{G}$; $f_t(g) = V_t^\pi(s,g) - V_{t-1}^\pi(s,g)$.

Hence, based on the insights from equation 4 and 8, we define the curriculum learning encoding as a policy confidence score $\mathcal{C}_t^\pi$ for a goal g present in the goal space $\mathcal{G}$ at timestep t:

$$f_t(g) = \mathcal{C}_t^\pi(g) = Q_t^\pi(s,g,a) \tag{9}$$

Next, we define curriculum based on the learning progress. The simplest strategy to measure the learning progress is to evaluate the temporal difference between the policy confidence score separated by Δt. This essentially captures the change in $\mathcal{C}^\pi$ for each goal g in goal space $\mathcal{G}$ separated by Δt time-steps. However, state-action value estimates are often noisy (Libardi et al., 2021; Raileanu & Fergus, 2021); Hence, this strategy fails to convey the relative importance of change in the policy confidence score.

We address this issue using the temporal variance in the policy confidence score as a metric to evaluate the learning progress. The temporal divergence provides a more meaningful and accurate performance measure by evaluating the policy over a time horizon rather than evaluating performance at a single time step. This temporal evolution reduces the noise effect and provides a more robust learning progress measure by inherently capturing information on both policy and Q-value evolution, as highlighted in equation 4. This temporal divergence-based learning progress can be formalized as follows.

Given a goal $g \in \mathcal{G}$, let $\mathcal{C}_t^\pi(g)$ denote the policy confidence score at time step t for goal g. We define the temporal divergence $\delta_{\mathcal{C}_t^\pi}(g)$ as the standard deviation of the policy confidence score across the last n time steps. Based on this, we define the learning progress score for goal g at time step t as:

$$LP^\pi(g,t) = \frac{1}{Z}\delta_{\mathcal{C}_t^\pi}(g), \tag{10}$$

where Z is the normalization constant given by the integral over the goal space $\mathcal{G}$:

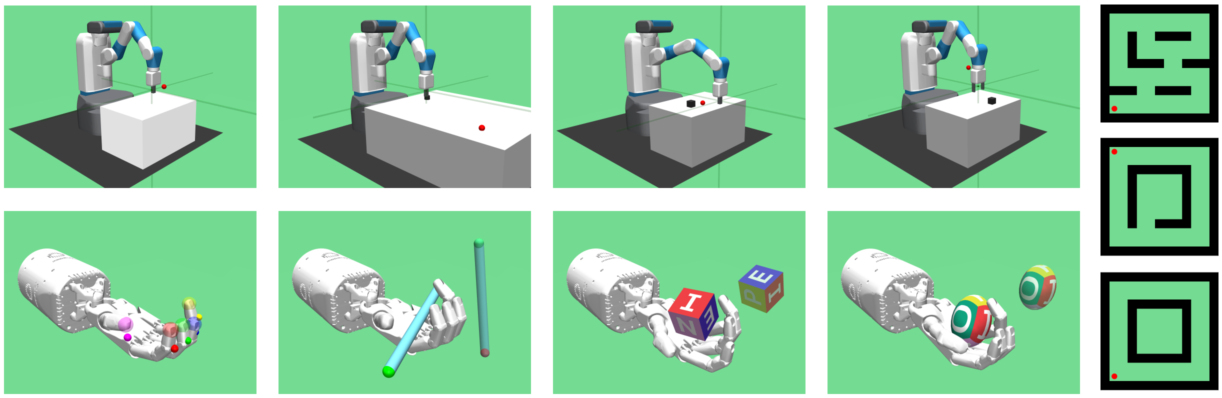

Figure 1: We evaluate the performance of TEACH on 4 FetchArm, 4 HandManipulate OpenAI Gymnasium (Plappert et al., 2018) environments, and 3 Maze navigation environments taken from (Zhang et al., 2020).

$$Z = \int_{\mathcal{G}} \delta_{\mathcal{C}_t^{\pi}}(g)\, dg. \tag{11}$$

This approach can be interpreted as a maximization problem, where the curriculum prioritizes goals based on their associated learning progress scores. The objective is to guide the agent's learning by focusing on the goals that lead to the most significant evolution of the policy, as indicated by the temporal divergence. This sampling strategy provides a way to represent and evaluate learning progress and seeks to prioritize goals with the highest learning progress, where the temporal divergence quantifies the learning progress. Since the goal space $\mathcal{G}$ is continuous to make the problem intractable, we uniformly sample N goals from the goal space.

4.4 Convergence Analysis

Theorem 1: Let $\mathcal{G}$ be the set of goals in the goals space and the agent's Q-function for goal $g \in \mathcal{G}$ at time t be denoted as $Q_t^{\pi}(s, a, g)$. Suppose the curriculum expands based on the variance of Q-function estimates over the past n time steps. For a fixed threshold $\eta > 0$, a goal $g \in \mathcal{G}$ is considered stable when $\mathrm{Var}_t(g) \leq \eta$, where $\mathrm{Var}_t(g)$ is the variance of Q-function estimates over n time steps.

If for all goals $g \in \mathcal{G}$, $\mathrm{Var}_t(g) \to 0$ as $t \to \infty$, then:

- The curriculum will eventually include all goals in $\mathcal{G}$.
- The agent's policy π_θ will converge to the optimal goal-conditioned policy π_θ^*, which satisfies:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{h=0}^{H-1} \gamma^h \mathcal{R}(s_h, a_h, g) \,\middle|\, s_0, g\right], \tag{12}$$

Proof:

Value Function Convergence: As the agent learns each goal g, the Q-function estimate $Q_t^{\pi}(s, a, g)$ converges to the true Q-function $Q^*(s, a, g)$ for goal g. Specifically, we assume that the agent's Q-function estimates satisfy:

$$\lim_{t \to \infty} Q_t^{\pi}(s, a, g) = Q^*(s, a, g), \quad \forall g \in \mathcal{G}. \tag{13}$$

Table 1: Complexity of enviornments

Environment	Reward	State	Context	Action	Goal space size	Episode length
FetchReach-v2	binary	$\mathbb{R}^{10}$	$\mathbb{R}^3$	$\mathbb{R}^3$	$1e^3$	50
FetchPickAndPlace-v2	binary	$\mathbb{R}^{25}$	$\mathbb{R}^3$	$\mathbb{R}^4$	$1e^3$	50
FetchSlide-v2	binary	$\mathbb{R}^{25}$	$\mathbb{R}^3$	$\mathbb{R}^3$	$1e^3$	50
FetchPush-v2	binary	$\mathbb{R}^{25}$	$\mathbb{R}^3$	$\mathbb{R}^3$	$1e^3$	50
HandManipulateBlock-v1	binary	$\mathbb{R}^{61}$	$\mathbb{R}^7$	$\mathbb{R}^{20}$	$1e^3$	100
HandManipulatePen-v1	binary	$\mathbb{R}^{61}$	$\mathbb{R}^7$	$\mathbb{R}^{20}$	$1e^3$	100
HandManipulateEgg-v1	binary	$\mathbb{R}^{61}$	$\mathbb{R}^7$	$\mathbb{R}^{20}$	$1e^3$	100
HandReach-v1	binary	$\mathbb{R}^{63}$	$\mathbb{R}^{15}$	$\mathbb{R}^{20}$	$1e^3$	50
MazeA-v0	binary	$\mathbb{R}^2$	$\mathbb{R}^2$	$\mathbb{R}^2$	$1e^3$	50
MazeB-v0	binary	$\mathbb{R}^2$	$\mathbb{R}^2$	$\mathbb{R}^2$	$1e^3$	50
MazeC-v0	binary	$\mathbb{R}^2$	$\mathbb{R}^2$	$\mathbb{R}^2$	$1e^3$	50

Consequently, for large t, the Q-function estimates $Q_t^\pi(s,a,g)$ will no longer vary significantly over time. This implies that the mean Q-function estimate $(\overline{Q}_t^\pi(g))$ over the past n step as $t\to\infty$ follows; $\lim_{t\to\infty}\overline{Q}_t^\pi(g) = Q^*(s,a,g)$. As a result, the variance $\mathrm{Var}_t(g)$ will approach zero:

$$\lim_{t\to\infty}\mathrm{Var}_t(g) = 0. \tag{14}$$

Curriculum Expansion Criterion: The curriculum expands only when the Q-function variance for all goals in the current curriculum $\mathcal{G}_{\text{curr}}$ satisfies the condition:

$$\mathrm{Var}_t(g) \le \eta, \quad \forall g \in \mathcal{G}_{\text{curr}}. \tag{15}$$

Since $\mathrm{Var}_t(g) \to 0$ as $t\to\infty$, there exists a time step t_1 such that for all goals $g\in\mathcal{G}_{\text{curr}}$, the variance $\mathrm{Var}_t(g)$ is smaller than or equal to some small threshold η, and Once this condition is satisfied, a new goal $g' \in \mathcal{G}\setminus\mathcal{G}_{\text{curr}}$ is added to the curriculum.

$$\mathrm{Var}_t(g) \le \eta, \quad \forall g \in \mathcal{G}_{\text{curr}}, \quad \text{for } t \ge t_1. \tag{16}$$

Inductive Step and Final Convergence: After adding a new goal g' to the curriculum, the agent starts learning the Q-function $Q_t^\pi(s,a,g')$ for the new goal. Initially, the variance $\mathrm{Var}_t(g')$ will likely be large, reflecting the agent's initial uncertainty. However, as the agent continues learning, the Q-function $Q_t^\pi(s,a,g')$ will converge to $Q^*(s,a,g')$, and the variance $\mathrm{Var}_t(g')$ will decrease over time.

Therefore, for each goal $g\in\mathcal{G}$, as time progresses, its variance will eventually reach the threshold η, and the goal will be considered learned. Since the variance for each goal decreases over time and the curriculum only expands when all current goals are sufficiently learned (i.e., their Q-function variance is below the threshold η), the curriculum will eventually contain all goals in $\mathcal{G}$. Once all goals have been added, the agent will have learned the Q-function for all goals, and its policy will converge to the optimal goal-conditioned policy π_θ^*, which maximizes the expected return for each goal in $\mathcal{G}$. Hence, the agent's policy converges to the optimal goal-conditioned policy, and the curriculum will eventually include all goals in $\mathcal{G}$.

Finally, we summarize the proposed method in Algorithm 2. Our approach employs the Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015) algorithm as the base RL policy. To enhance the efficiency of universal policy learning, we incorporate the Hindsight Experience Replay (HER) (Andrychowicz et al., 2017) strategy into the replay buffer manipulation. HER allows the off-policy RL algorithm to learn more effectively by reinterpreting unsuccessful trajectories as successful.

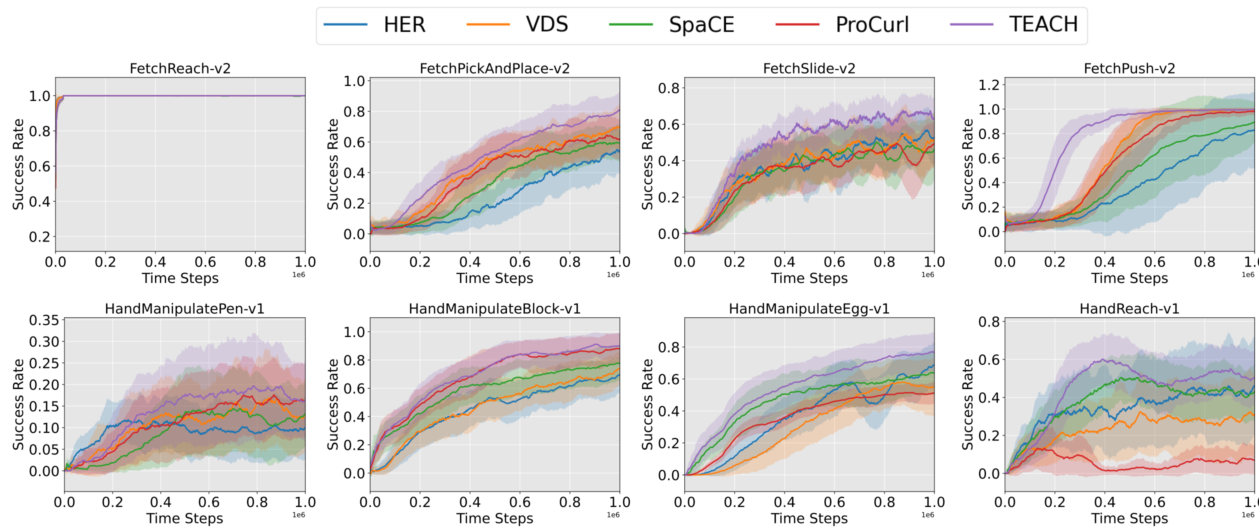

Figure 2: Results show the performance across 8 robotic manipulation tasks (refer supplementary for complete results). The plots show the success rate along the y-axis evaluated through current policy. The reported results are mean across 5 seeds with shaded regions highlighting standard deviation.

A notable advantage of our method is that the teacher component, responsible for curriculum learning, does not introduce any new hyperparameters except temporal window (n), which captures the evolution of the policy confidence score over past n time steps. This design choice ensures that the training routine remains consistent with the base RL policy, simplifying the implementation and reducing the potential for hyperparameter tuning complexity. As a result, our method maintains the stability and reliability of the base RL algorithm while enhancing its learning efficiency through strategic replay buffer manipulation and curriculum learning.

5 Experiments

We test our method on 11 multi-goal binary reward tasks. The complexity details of the task are shown in Table 1, context refers to the goal's dimension, and goal space size (N) refers to the number of goals sampled from continuous goal space to make the problem tractable. While our main focus relies on the robotics task environment (Plappert et al., 2018), we also include three Maze navigation tasks (Zhang et al., 2020) to evaluate performance in low contextual problems. Refer to Appendix B for task definitions.

5.1 Baselines

To establish the contribution and effectiveness of the proposed approach, we compare our method with the following baselines. The VDS and ProCurl are currently state-of-the-art in multi-goal and multi-task settings, respectively. All the baselines use value estimates to design ACL, which makes them a strong choice for baselines to highlight the advantage of our approach.

HER-IID (Andrychowicz et al., 2017) In HER-IID, the RL agent uses a hindsight experience reply buffer which independently and identically samples goals from task space. We use the official code-based implementation to reproduce the results.

VDS (Zhang et al., 2020) Value Disagreement Sampling samples goals from the goal space based on value disagreement. Their strategy prioritizes goals that maximize the epistemic uncertainty of the Q-function of the policy. We use the official implementation to reproduce the results.

SPaCE (Eimer et al., 2021) Self-Paced Context Evaluation provides a curriculum learning explicitly using agent's performance as an ordering criterion. They use the agent's state value predictions to generate

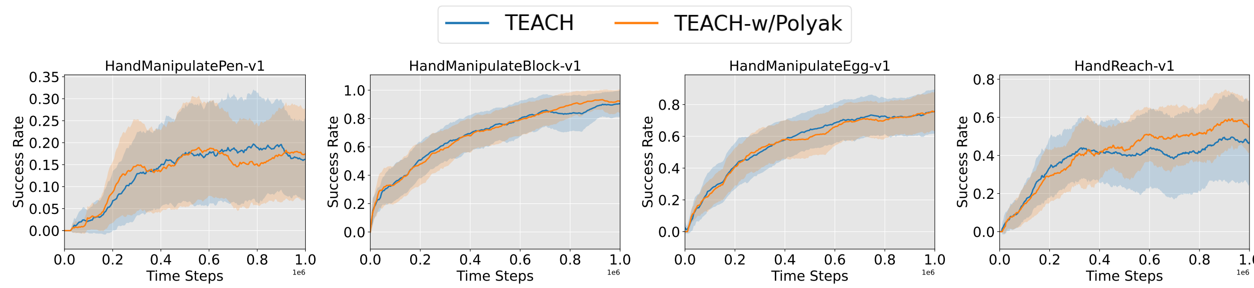

Figure 3: Results show the effect of smooth target confidence score for temporal window size $n = 3$. The results are across 5 seeds where the shaded region represents the standard deviation (refer to Figure 7 for complete results). We observe that the smooth target confidence score compared to the standard confidence score led to similar performances. This validates the hypothesis that the proposed method measure is robust to noisy value estimates.

curriculum. Which uses the temporal difference to design a curriculum that heuristically selects a new goal if it observes significant learning in the value function. We extended their strategy to our setting using DDPG+HER (Andrychowicz et al., 2017) implementation.

ProCuRL (Tzannetos et al., 2023) Proximal Curriculum for Reinforcement Learning Agents is inspired by the pedagogical concept of the zone of proximal development. They capture the idea of proximal using state value estimates w.r.t. the learner's current policy. Their strategy essentially serves as a geometric mean of value function over goals and sample goal with the highest geometric mean. We extended their strategy to our setting using DDPG+HER (Andrychowicz et al., 2017) implementation.

5.2 Implementation Details

Our curriculum design comprises two primary components: the teacher (goal proposer) and the student (RL policy). The teacher is a non-learning module that evaluates the student's capabilities (the RL policy) to propose goals that promote efficient learning. The RL policy, parameterized by θ, is deterministic by design, and we add noise to its actions to enable better exploration. We combine DDPG with HER. Incorporating the HER replay buffer relabeling strategy enhances learning efficiency.

At the beginning of each episode, the student queries the teacher for a target goal. The teacher evaluates the change in the student's learning progress for each goal within the goal space, which consists of N goals. The teacher selects and assigns the goal with maximum temporal uncertainty in the student's confidence score as the target goal. The RL policy then focuses on achieving this assigned goal. Transitions generated through agent-environment interactions are stored in the replay buffer and subsequently used to update the policy. The network architecture is MLP with 2 layers for Fetch and Maze navigation tasks and 3 layers for the HandManipulation task, respectively, for both the actor and critic networks. The networks are trained using a learning rate of 0.001 with a batch size of 1024; refer to supplementary for hyperparameter details. All robotics tasks are trained for 1M time steps and maze navigation tasks for 400K time steps, respectively. The agent's performance is evaluated by randomly sampling goals from the goal space. All the reported results highlight the agent's success rate in achieving those randomly sampled goals. Specifically, the success rate is calculated as the average success rate in goal accomplishment, which averaged over 20 episodes (each with a randomly sampled goal) at each evaluation step.

5.3 Improvement Through TEACH

The performance of all methods on robotics manipulation tasks is shown in Figure 1. These results demonstrate that our proposed approach, which inherently combines dual exploitation of the information from the current policy and state-action value estimates through temporal evolution of Q-value, achieves superior sample efficiency. The performance improvements are evident across low-context tasks, such as *Maze naviga-*

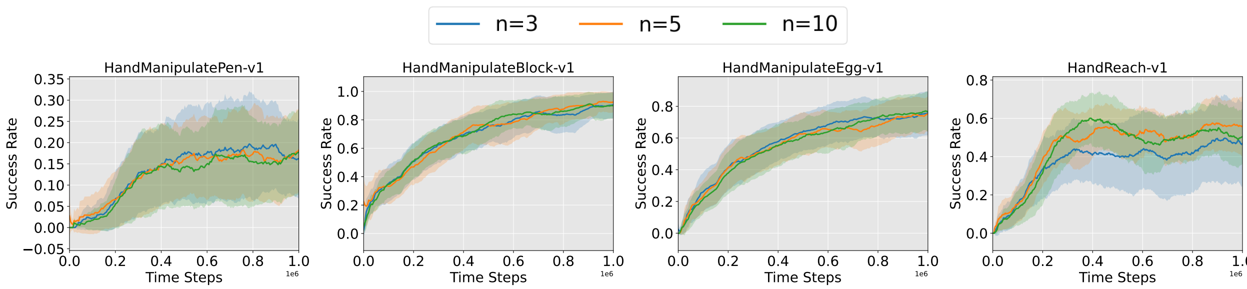

Figure 4: Ablation analysis of the effect of temporal window size n. The results are across 5 seeds where the shaded region represents the standard deviation (refer to Figure 8 for complete results). We observe that the presented approach is insensitive to temporal window size.

tion, and high-context tasks, such as *HandManipulation*. The reported performance of TEACH is computed using a temporal window of $n = 10$, i.e., the curriculum is designed based on the temporal divergence of the Q-function over the past 10 time steps. However, we highlight in Section 5.5 that the performance of TEACH is mostly invariant with the size of the temporal history of the value function.

Our approach consistently yields better policies while maintaining high sample efficiency, as evidenced by extensive comparisons across tasks and baselines. The improvements stem from addressing two primary challenges: (1) the noisy and often inaccurate nature of value estimates and (2) the additional noise introduced by the HER relabeling strategy, which adds fictitious and biased data into the replay buffer used for training the value function.

We mitigate these challenges by designing a curriculum that better aligns with the current policy's capabilities and is less prone to the detrimental effects of noisy value estimates. By leveraging dual exploitation—drawing information from both the current policy and state-action value estimates using temporal divergence to measure learning progress, our strategy reduces the impact of noise and enhances the robustness and effectiveness of policy learning.

5.4 Effect of Smooth Target Confidence Score on Performance

The issue of noisy value estimates and bias can be addressed by Polyak averaging (Polyak & Juditsky, 1992) to obtain a smooth value function. The smoothed values function is often called the 'target' values function. The target function parameters are computed using equation 17, where $\theta^{'}$ are parameters of the target values function and θ are parameters of the value function.

$$\theta^{'} = \alpha\theta^{'} + (1 - \alpha)\theta \tag{17}$$

We conduct an ablation study incorporating a smooth state-action value function to evaluate the impact of a smooth confidence score. The results, presented in Figure 3, indicate that a smooth target state-action value function does not improve performance compared to the noisy state-action values function used in the proposed algorithm. The smooth success score provides better results in *FetchPickAndPlace* task but performs poorly for *FetchPush* task while performing similarly across the remaining tasks. This underscores the ability of the TEACH to be robust to noisy value estimates, which further validates the arguments made in the section 4. The impact of the noisy value estimates can be reduced further to some extent by using a larger temporal window size.

5.5 Effect of Size of Temporal Window

In this analysis, we investigate the effect of the temporal window size n on our proposed automatic curriculum learning method, TEACH. As shown in Figure 4, the proposed approach demonstrates a high degree of invariance to the size of the temporal window n, which captures the length of past time steps used for computing temporal divergence of policy confidence score. This invariance highlights the robustness of

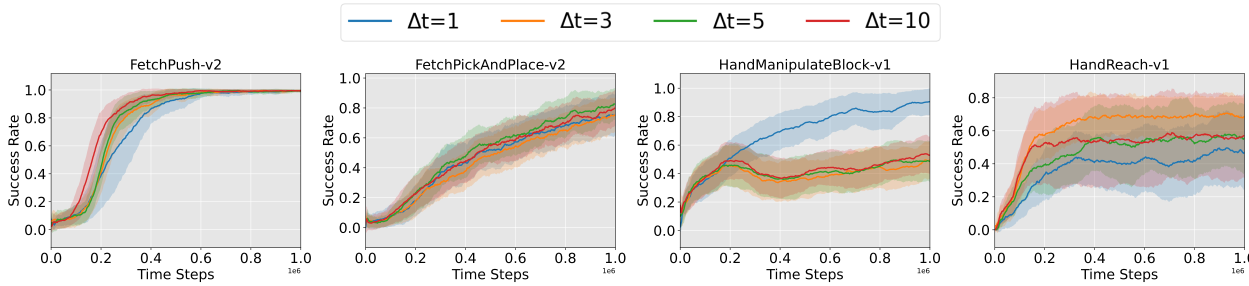

Figure 5: Ablation analysis of the effect of student-teacher interplay frequency Δt. The results are across 5 seeds where the shaded region represents the standard deviation. We observe that the performance can be improved using different interplay frequencies for different tasks.

TEACH in adapting to different temporal contexts, making it suitable for diverse task settings. However, while the overall performance remains consistent across different values of n, we observe that $n = 10$ produces slightly better results in some tasks. This may be because $n = 10$ strikes an optimal balance between capturing sufficient historical information and avoiding sensitivity to noise compared to smaller temporal windows. Therefore, we report results using $n = 10$ in Figure 1 for consistency and reproducibility.

5.6 Effect of Teacher-Student Interplay Frequency

In this ablation study, we investigate the effect of the teacher-student interplay frequency (Δt) on learning performance. The interplay frequency controls the temporal interval, after which changes in the policy confidence score are evaluated. As illustrated in Figure 5, a less aggressive interplay frequency can lead to faster learning. These results suggest that a continuous curriculum generation strategy may induce forgetting behaviors, as agents are exposed to diverse goals at varying time steps. Conversely, oversampling specific goals allows agents to focus on a consistent set of objectives, thereby facilitating improved learning. However, our findings reveal that achieving more stable and consistent performance requires an adaptive interplay approach, yielding superior results to fixed strategies, which may require extensive fine-tuning. While this work employs a fixed teacher-student interplay frequency ($\Delta t = 1$) to underscore the benefits of a temporal divergence-driven curriculum strategy. Nevertheless, we emphasize that the design of adaptive curriculum strategies that dynamically adjust the interplay between the teacher and the student to optimize learning performance can result in superior performance. Notably, prior works have not addressed the importance of adaptability in curriculum generation, and we believe that future research should explore the design of adaptive curriculum strategies that dynamically adjust the interplay between the teacher and the student to optimize learning performance.

6 Conclusion

In this work, we introduced a novel curriculum strategy for goal-conditioned reinforcement learning that leverages a temporal divergence-based approach to address the challenges posed by noisy value estimates in curriculum design. We first demonstrated theoretically how temporal divergence serves as an upper bound on policy divergence, providing a more reliable mechanism for evaluating both policy performance and the value function. Building on this foundation, we provided a convergence analysis. Subsequently, we conducted extensive experiments across 11 robotics and navigation environments with binary rewards, showcasing the practical effectiveness of our strategy. In these experiments, we highlighted the robustness of our algorithm, demonstrating its insensitivity to smooth target success score estimates and its ability to mitigate the effects of noisy value estimates. Finally, we demonstrated that an adaptive curriculum learning strategy, built upon our temporal divergence-based framework, can outperform fixed strategies, yielding improved and more stable results.

References

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Per-Arne Andersen, Morten Goodwin, and Ole-Christoffer Granmo. Deep rts: a game environment for deep reinforcement learning in real-time strategy games. In *2018 IEEE conference on computational intelligence and games (CIG)*, pp. 1–8. IEEE, 2018.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Richard Bellman. Dynamic programming. *science*, 153(3731):34–37, 1966.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning (icml). *Google Scholar Google Scholar Digital Library Digital Library*, 2009.

Angelo Cangelosi and Matthew Schlesinger. *Developmental robotics: From babies to robots*. MIT press, 2015.

Gaurav Chaudhary, Laxmidhar Behera, and Tushar Sandhan. Active perception system for enhanced visual signal recovery using deep reinforcement learning. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10097084.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pp. 2048–2056. PMLR, 2020.

Mehul Damani and Lerrel Pinto. Continuous goal sampling: A simple technique to accelerate automatic curriculum learning, 2023. URL `https://openreview.net/forum?id=Vk9RH9aL1Yv`.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

Theresa Eimer, André Biedenkapp, Frank Hutter, and Marius Lindauer. Self-paced context evaluation for contextual reinforcement learning. In *International Conference on Machine Learning*, pp. 2948–2958. PMLR, 2021.

Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993.

Meng Fang, Tianyi Zhou, Yali Du, Lei Han, and Zhengyou Zhang. Curriculum-guided hindsight experience replay. *Advances in neural information processing systems*, 32, 2019.

Yannis Flet-Berliac and Philippe Preux. Samples are useful? not always: denoising policy gradient updates using variance explained. 2019.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on robot learning*, pp. 482–495. PMLR, 2017.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *International conference on machine learning*, pp. 1515–1528. PMLR, 2018.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *international conference on machine learning*, pp. 1311–1320. Pmlr, 2017.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

Dong Han, Beni Mulyana, Vladimir Stankovic, and Samuel Cheng. A survey on deep reinforcement learning algorithms for robotic manipulation. *Sensors*, 23(7):3762, 2023.

Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*, 2018.

Edward S Hu, Richard Chang, Oleh Rybkin, and Dinesh Jayaraman. Planning goals for exploration. *arXiv preprint arXiv:2303.13002*, 2023.

Minqi Jiang, Michael Dennis, Jack Parker-Holder, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Replay-guided adversarial environment design. *Advances in Neural Information Processing Systems*, 34:1884–1897, 2021a.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning*, pp. 4940–4950. PMLR, 2021b.

Avi Karni, Gundela Meyer, Christine Rey-Hipolito, Peter Jezzard, Michelle M Adams, Robert Turner, and Leslie G Ungerleider. The acquisition of skilled motor performance: fast and slow experience-driven changes in primary motor cortex. *Proceedings of the National Academy of Sciences*, 95(3):861–868, 1998.

Elia Kaufmann, Antonio Loquercio, Rene Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Deep drone racing: Learning agile flight in dynamic environments. In *Conference on Robot Learning*, pp. 133–145. PMLR, 2018.

Pascal Klink, Hany Abdulsamad, Boris Belousov, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. A probabilistic interpretation of self-paced learning with applications to reinforcement learning. *Journal of Machine Learning Research*, 22(182):1–52, 2021.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Qiyang Li, Yuexiang Zhai, Yi Ma, and Sergey Levine. Understanding the complexity gains of single-task rl with a curriculum. In *International Conference on Machine Learning*, pp. 20412–20451. PMLR, 2023.

Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

Gabriele Libardi, Gianni De Fabritiis, and Sebastian Dittert. Guided exploration with proximal policy optimization using a single demonstration. In *International Conference on Machine Learning*, pp. 6611–6620. PMLR, 2021.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher–student curriculum learning. *IEEE transactions on neural networks and learning systems*, 31(9):3732–3740, 2019.

Russell Mendonca, Oleh Rybkin, Kostas Daniilidis, Danijar Hafner, and Deepak Pathak. Discovering and achieving goals via world models. *Advances in Neural Information Processing Systems*, 34:24379–24391, 2021.

Michèle Molina and François Jouen. Modulation of the palmar grasp behavior in neonates according to texture property. *Infant Behavior and Development*, 21(4):659–666, 1998.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pp. 2817–2826. PMLR, 2017.

Silviu Pitis, Harris Chan, Stephen Zhao, Bradly Stadie, and Jimmy Ba. Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. In *International Conference on Machine Learning*, pp. 7750–7761. PMLR, 2020.

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

Vitchyr H Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*, 2019.

Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In *Conference on Robot Learning*, pp. 835–853. PMLR, 2020a.

Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. Automatic curriculum learning for deep rl: A short survey. *arXiv preprint arXiv:2003.04664*, 2020b.

Sebastien Racaniere, Andrew K Lampinen, Adam Santoro, David P Reichert, Vlad Firoiu, and Timothy P Lillicrap. Automated curricula through setter-solver interactions. *arXiv preprint arXiv:1909.12892*, 2019.

Roberta Raileanu and Rob Fergus. Decoupling value and policy for generalization in reinforcement learning. In *International Conference on Machine Learning*, pp. 8787–8798. PMLR, 2021.

Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. Epopt: Learning robust neural network policies using model ensembles. *arXiv preprint arXiv:1610.01283*, 2016.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pp. 1312–1320. PMLR, 2015.

Oliver G Selfridge, Richard S Sutton, and Andrew G Barto. Training and tracking in robotics. In *Ijcai*, pp. 670–672, 1985.

Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov, Gabriel Synnaeve, Arthur Szlam, and Rob Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*, 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, et al. Open-ended learning leads to generally capable agents. *arXiv preprint arXiv:2107.12808*, 2021.

Georgios Tzannetos, Bárbara Gomes Ribeiro, Parameswaran Kamalaruban, and Adish Singla. Proximal curriculum for reinforcement learning agents. *arXiv preprint arXiv:2304.12877*, 2023.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Lev Semenovich Vygotsky and Michael Cole. *Mind in society: Development of higher psychological processes*. Harvard university press, 1978.

Daphna Weinshall, Gad Cohen, and Dan Amir. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International Conference on Machine Learning*, pp. 5238–5246. PMLR, 2018.

Gaurav Yengera, Rati Devidze, Parameswaran Kamalaruban, and Adish Singla. Curriculum design for teaching via demonstrations: Theory and applications. *Advances in Neural Information Processing Systems*, 34:10496–10509, 2021.

Yunzhi Zhang, Pieter Abbeel, and Lerrel Pinto. Automatic curriculum learning through value disagreement. *Advances in Neural Information Processing Systems*, 33:7648–7659, 2020.

A Correlation Between Q-Value and Policy Divergence

We begin by considering a soft policy update mechanism (Haarnoja et al., 2017), where the policy update rule can be defined as:

$$\pi_{\theta_t}(a|s) \propto \exp\left(\frac{Q_t^\pi(s,a)}{\alpha}\right), \tag{18}$$

where $\alpha > 0$ is a temperature parameter that governs the trade-off between exploration and exploitation. The partition function $Z_t(s) = \sum_a \exp(Q_t^\pi(s,a)/\alpha)$ normalizes the policy distribution $\pi_{\theta_t}(a|s)$.

To understand the changes in the policy across consecutive updates, we analyze the Kullback-Leibler (KL) divergence (Kullback & Leibler, 1951) between the two policies:

$$\mathrm{KL}(\pi_{\theta_{t+1}} \parallel \pi_{\theta_t}) = \mathbb{E}_s\left[\sum_a \pi_{\theta_{t+1}}(a|s) \log \frac{\pi_{\theta_{t+1}}(a|s)}{\pi_{\theta_t}(a|s)}\right]. \tag{19}$$

By expressing the policy in the softmax form $\pi_{\theta_t}(a|s) = \frac{\exp(Q_t^\pi(s,a)/\alpha)}{Z_t(s)}$, the KL divergence expands as:

$$\mathrm{KL}(\pi_{\theta_{t+1}} \parallel \pi_{\theta_t}) = \mathbb{E}_s\left[\sum_a \pi_{\theta_{t+1}}(a|s)\left(\frac{Q_{t+1}^\pi(s,a) - Q_t^\pi(s,a)}{\alpha}\right) - \log\frac{Z_{t+1}(s)}{Z_t(s)}\right]. \tag{20}$$

Now, recall that the partition functions are given by:

$$Z_t(s) = \sum_a \exp(Q_t^\pi(s,a)/\alpha), \quad Z_{t+1}(s) = \sum_a \exp(Q_{t+1}^\pi(s,a)/\alpha), \tag{21}$$

we rewrite $Z_{t+1}(s)$ in terms of $Z_t(s)$ and π_{θ_t} by using a small Q-value and policy update approximation. The ratio of partition functions becomes:

$$\frac{Z_{t+1}(s)}{Z_t(s)} = \mathbb{E}_{a\sim\pi_{\theta_t}}\left[e^{\frac{\Delta Q(s,a)}{\alpha}}\right], \tag{22}$$

where $\Delta Q(s,a) = Q_{t+1}^\pi(s,a) - Q_t^\pi(s,a)$.

To simplify further, we expand the ratio of partition functions for small policy updates using a Taylor series expansion:

$$\frac{Z_{t+1}(s)}{z_t(s)} = \mathbb{E}_{a\sim\pi_{\theta_t}}\left[e^{\frac{\Delta Q(s,a)}{\alpha}}\right] \approx 1 + \frac{1}{\alpha}\mathbb{E}_{a\sim\pi_{\theta_t}}[\Delta Q(s,a)] + \frac{1}{2\alpha^2}\mathbb{E}_{a\sim\pi_{\theta_t}}[(\Delta Q(s,a))^2]. \tag{23}$$

Next, taking the logarithm and using the approximation $\log(1+x) \approx x - \frac{x^2}{2}$ for small x, after simplification we obtain:

$$\log\frac{Z_{t+1}(s)}{Z_t(s)} \approx \frac{1}{\alpha}\,\mathbb{E}_{a\sim\pi_{\theta_t}}[\Delta Q(s,a)] + \frac{1}{2\alpha^2}\,\mathrm{Var}_{a\sim\pi_{\theta_t}}(\Delta Q(s,a)). \tag{24}$$

Now, using this approximation, the KL divergence can be expressed as:

$$\mathrm{KL}(\pi_{\theta_{t+1}} \parallel \pi_{\theta_t}) \approx \mathbb{E}_s\left[\frac{1}{\alpha}\left(\mathbb{E}_{a\sim\pi_{\theta_{t+1}}}[\Delta Q(s,a)] - \mathbb{E}_{a\sim\pi_{\theta_t}}[\Delta Q(s,a)]\right) - \frac{1}{2\alpha^2}\,\mathrm{Var}_{a\sim\pi_{\theta_t}}(\Delta Q(s,a))\right]. \tag{25}$$

Finally, under the assumption of small policy updates ($\pi_{\theta_{t+1}} \approx \pi_{\theta_t}$) using first-order policy approximation:

$$\mathbb{E}_{a\sim\pi_{\theta_t}}[\Delta Q] \approx \mathbb{E}_{a\sim\pi_{\theta_t}}[\Delta Q] + \frac{1}{\alpha}\,\mathrm{Var}_{a\sim\pi_{\theta_t}}(\Delta Q), \tag{26}$$

Hence, the KL divergence between two successive policy updates can be approximated as:

$$\mathrm{KL}(\pi_{\theta_{t+1}} \parallel \pi_{\theta_t}) \approx \frac{1}{2\alpha^2}\mathbb{E}_s\left[\mathrm{Var}_{a\sim\pi_{\theta_t}}(\Delta Q(s,a))\right]. \tag{27}$$

B Task Definition

FetchReach: Move the gripper to a target location.

FetchPickAndPlace: Pick up a block and place it at a target location.

FetchPush: Push the clock to the desired position.

FetchSlide: Slide the block to a position outside the robotic arm workspace.

HandManipulateBlock: Rotate the block to reach the target rotation in the z-axis.

HandManipulatePen: Rotate the pen to reach the target rotation in all axes.

HandManipulateEgg: Rotate the egg to reach the target rotation in all axes.

HandReach: Move to match a target position for each fingertip.

Maze: The environment for navigation tasks is a finite-sized, 2-dimensional maze with blocks. The agent is given a target position and starts from a fixed point in the maze, and it obtains a reward of 0 if it gets sufficiently close to the target position at the current time step or a penalty of -1 otherwise. The agent observes the 2-D coordinates of the maze, and the bounded action space is specified by velocity and direction. The agent moves along the direction with the velocity specified by the action if the new position is not a block and stays otherwise. The maximum time step of an episode is set to 50. MazeA, MazeB, and MazeC variants are shown in Figure 1.

C Hyperparameters

Table 2: Hyperparameters for TEACH

Hyperparameter	**Value**	**Description**
Actor Learning Rate (actor_lr)	1×10^{-3}	Learning rate for the actor-network
Critic Learning Rate (critic_lr)	1×10^{-3}	Learning rate for the critic-network
Layer Size (hidden)	512	Number of neuron in each hidden layer
Batch Size (batch_size)	1024	Number of transitions sampled per update step
Discount Factor (γ)	0.95	Discount factor for future rewards
Replay Buffer Size (buffer_size)	1×10^6	Maximum number of transitions in the buffer
Target Update Rate (τ)	0.005	Update rate for the target networks
Noise Scale (action_noise)	0.2	Initial scale for Ornstein-Uhlenbeck noise
Polyak Averaging Coefficient (polyak)	0.95	Interpolation factor for target network updates
HER Samples per Transition (future_k)	4	Number of hindsight samples per real transition
Goal Selection Strategy	Future	Future state is used as the hindsight goal
Warmup Steps	1000	Number of initial random steps before training
Training Frequency (train_freq)	1	Train the policy after every step
Gradient Clipping	1.0	Maximum gradient norm to stabilize training
Evaluation Episodes	20	Episodes used for evaluation every epoch

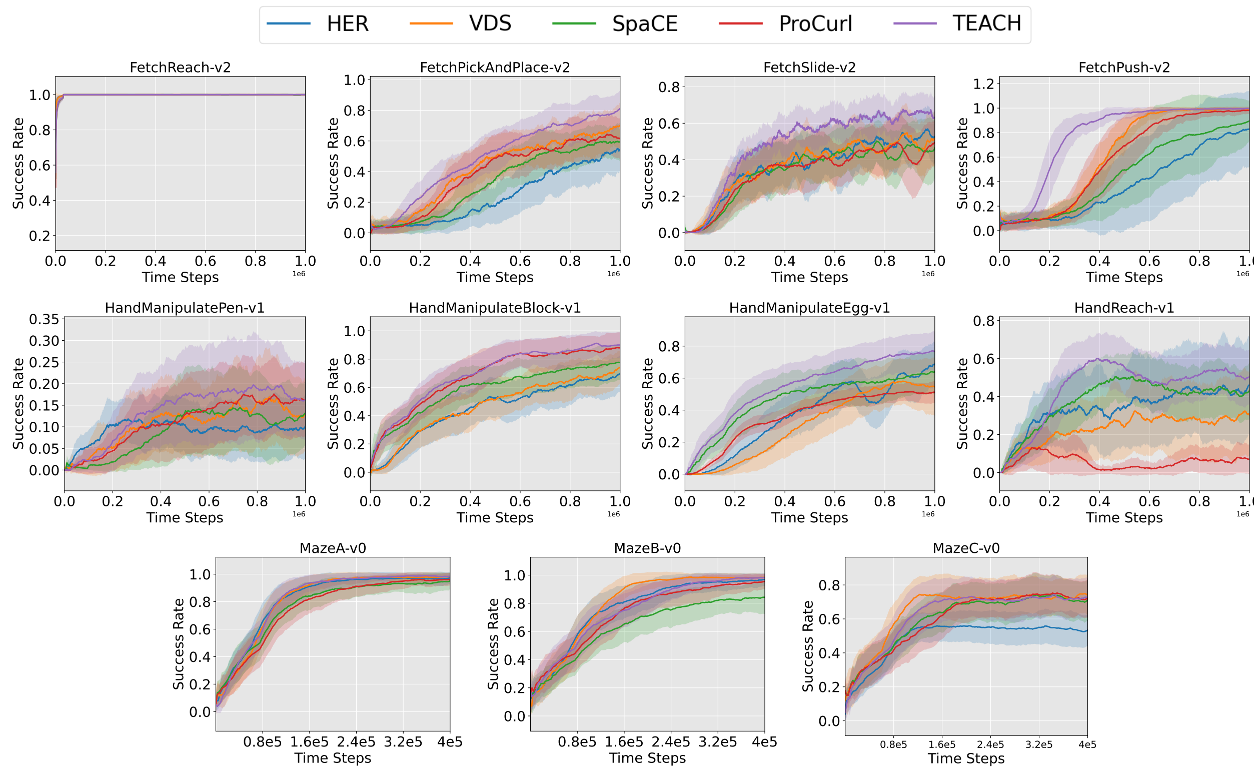

Figure 6: Results show the performance across 8 robotic manipulation tasks and 3 maze navigation tasks. The plots show the success rate along the y-axis evaluated through current policy. The reported results are mean across 5 seeds with shaded regions highlighting standard deviation.

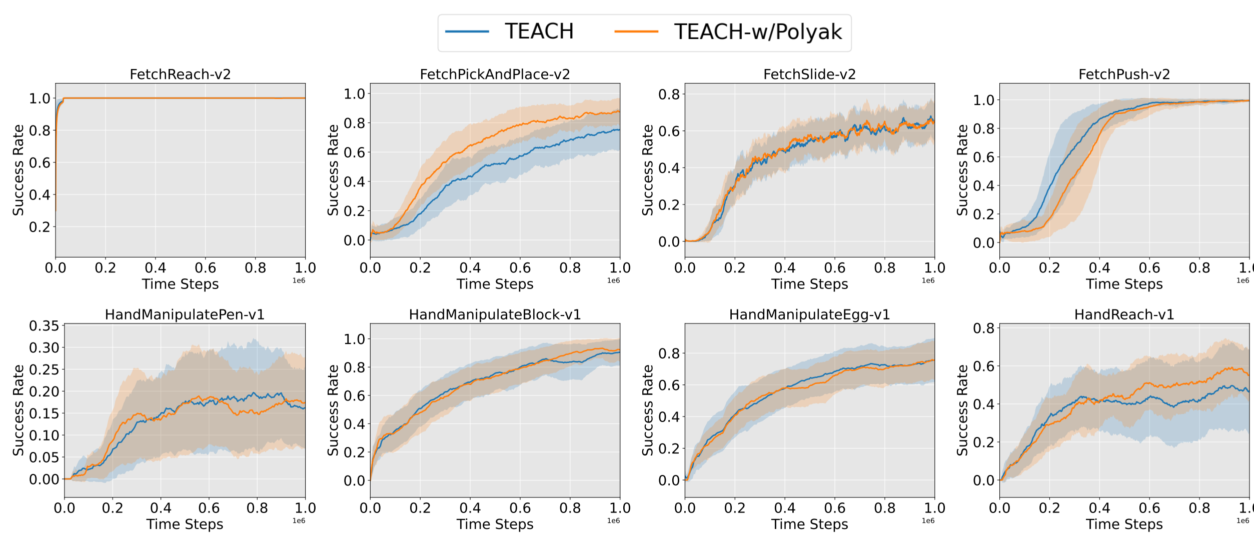

Figure 7: Results show the effect of smooth target confidence score for temporal window size $n = 3$. The results are across 5 seeds where the shaded region represents the standard deviation. We observe that the smooth target confidence score compared to the standard confidence score led to similar performances. This validates the hypothesis that the proposed method measure is robust to noisy value estimates.

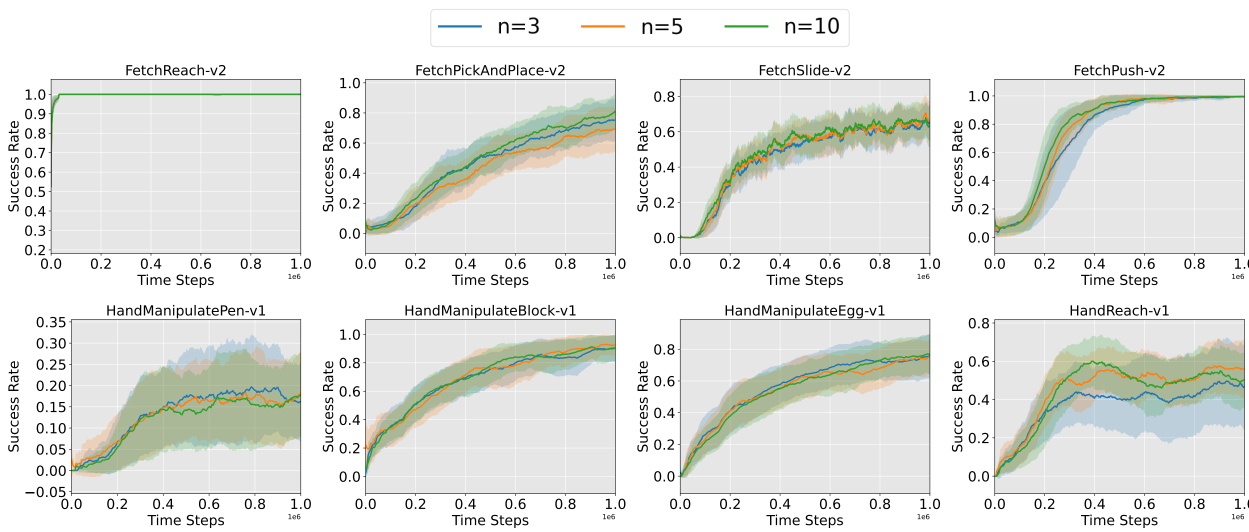

Figure 8: Ablation analysis of the effect of temporal window size n. The results are across 5 seeds where the shaded region represents the standard deviation. We observe that the presented approach is insensitive to temporal window size.