Contextual Situational Reinforcement Learning: Adapting to New Situation Dynamics with Context-aware Generalization and Off-dynamics Learning

Anonymous Authors[1]

Abstract

Situational Markov Decision Process (MDP) has been proposed to capture the dynamic and uncontrollable environmental factors that are crucial for decision making in real-world applications. However, in many cases, it is necessary to solve a set of situational MDPs that differ only in their situation dynamics, such as varying customer demands for inventory control at different warehouses. The main challenge here is to train an agent that can adapt to unseen environments without additional online training, potentially using offline situation data for the new environments. In this paper, we propose the Contextual Situational MDP, under which we address this challenge through *context-aware generalization* and *off-dynamics learning*. Context-aware generalization learns latent contexts to encode situation dynamics for different environments, allowing for generalization from training to test environments. Off-dynamics learning optimizes the policy directly on test environments with offline situation data, but without additional online training. Our experiments on locomotion tasks demonstrate that our algorithm can effectively adapt to different situation dynamics.

1. Introduction

In real-world applications, dynamic environmental factors, such as fluctuating user demand in inventory control or changing traffic patterns in autonomous driving, can significantly impact state transitions and rewards, but cannot be controlled by our agents. The problem has been formulated as Situational MDP (Chen et al., 2022), which is also known as Input-driven MDP (Mao et al., 2018b), or MDP with exogenous processes (Chitnis & Lozano-Pérez, 2020; Boutilier et al., 1999). The uncontrollable environmental factors are referred to as *situations* and their dynamics as *situation dynamics*. Researchers have focused on solving a single Situational MDP, by addressing challenges such as reducing variance (Mao et al., 2018a), separating situations and endogenous states from observations (Dietterich et al., 2018; Chitnis & Lozano-Pérez, 2020; Efroni et al., 2021; Pan et al., 2022) to learn more efficiently, and detecting the abrupt changes of latent situations (Chen et al., 2022).

However, in many real-world applications, we may need to solve a set of situational MDPs, which differ only in situation dynamics, but share the same state transitions and rewards. For example, a large company may have multiple warehouses in different locations with varying user demand patterns as different situation dynamics, but with the same logic on inventory replenishment and profits as the same state transitions and rewards. One possible approach is to treat these tasks [1] as different tasks and retrain a different policy for each, which could be time-consuming. Another approach is to treat these tasks as similar tasks and train one policy for all, which could be suboptimal due to not considering the differences between the tasks. The problem is, can we train an agent which can adapt to new situation dynamics without further online training on the new environment?

In this work, we propose a novel framework, the Contextual Situational Markov Decision Process (CS-MDP), for capturing a set of situational MDPs that differ only in their situational dynamics. Specifically, we introduce the concept of a latent super-context, represented by z, to capture the situational dynamics as $p_z(c' \mid c)$. At training time, we assume access to a diverse set of situational MDPs with good coverage over possible latent z. Additionally, we assume access to offline situation sequences for the test situational MDP with an unknown z. As the situation dynamics are independent of the agent's actions, access to offline situation sequences is commonly available in real-world applications, such as user demand sequences in inventory

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]We use the terms environment, task, and Situational MDP, interchangeably.

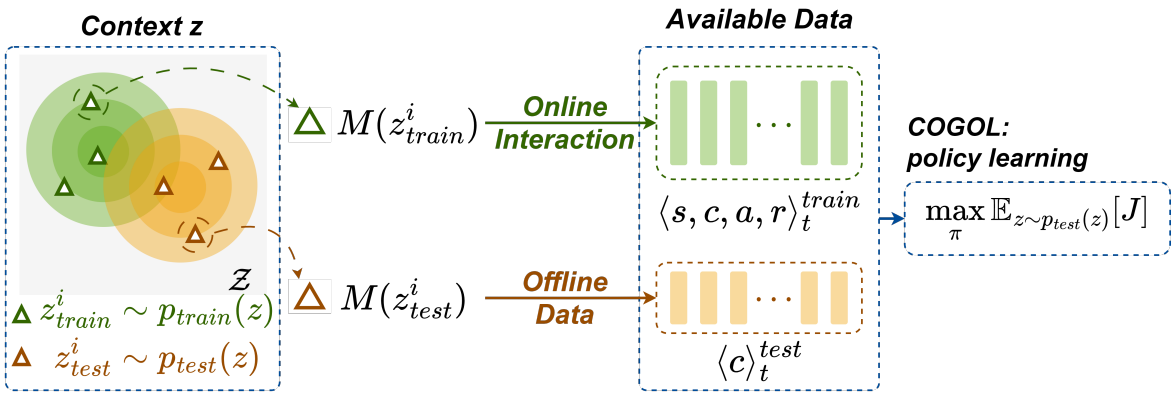

Figure 1. An overview of our problem setting. CS-MDP consists of a set of situational MDPs $M(z)$ with different contexts z. Samples of training contexts $z^i_{train} \sim p_{train}(z)$ is denoted in green triangles, and samples of test contexts $z^i_{test} \sim p_{test}(z)$ are denoted in orange triangles. The available data includes online interaction trajectories $\langle s, c, a, r \rangle_t^{train}$ on training environments $M(z^i_{train})$ and offline situation data $\langle c \rangle_t^{test}$ for test environments $M(z^i_{test})$. Our proposed algorithm COGOL can utilize these data to improve the expected performance on test environments.

control. The challenge here is training an agent capable of adapting to the new situation dynamics without additional online training on the new situational MDPs.

We address the challenge above by *context-aware generalization* and *off-dynamics learning*. The basic idea of context-aware generalization is to enable generalization from training to test environments, by conditioning policy on latent context which encodes the similarity between different situation dynamics of different environments. Specifically, we learn a context-dependent policy $\pi(\cdot|s_t, c_t, b_t)$ conditions on the inferred context belief b_t from recent situation dynamics and optimize the expected returns on training environments through online interactions.

Then, we propose off-dynamics learning to optimize the context-dependent policy for the test environment, by leveraging the offline situation sequences of the test environment and online interactions of the training environments. We present convergence results for off-dynamics learning of value functions on the test environment, supposing the contexts in training environments are informative enough with respect to the test one. Off-dynamics learning enables our context-dependent policy to be further optimized for the new environment, but without further online interactions.

To evaluate our algorithm, we conduct experiments on a 1D objective tracking environment with moving goals of multiple dynamics, as well as on MuJoCo tasks (Todorov et al., 2012) with varied situation dynamics. Experiment results demonstrate that our algorithm can adapt to different situation dynamics by generalizing over inferred context and off-dynamic learning on offline situation sequences. Our contributions can be listed as follows:

- We introduce Contextual Situational Markov Decision Process (CS-MDP), to capture a set of situational MDPs which differ only in situation dynamics.
- We propose COGOL, a deep RL algorithm that can adapt to new situation dynamics without further online training, by ***CO**ntext-aware **G**eneralization* and ***O**ff-dynamics **L**earning*.
- Experiments on locomotion tasks demonstrate that our algorithm can effectively adapt to new situation dynamics without further online training on new environments.

2. Problem Formulation

Situational MDP and Contextual Situational MDP We begin by reviewing the definition of the Situational MDP and then provide the formal definition of the Contextual Situational MDP (CS-MDP).

Definition 2.1. A *Situational MDP* is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{C}, p_c, p_s, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ is state space and action space respectively, $\mathcal{C}$ is the context/situation space, $p_c(c_{t+1} \mid c_t)$ is the situation transition function, $p_s(s_{t+1} \mid s_t, a_t, c_t)$ is the state transition function, $r(s_t, a_t, c_t)$ is the reward function, and $\gamma \in (0, 1]$ is the discount factor.

Building upon the definition of the Situational MDP, we formally define the CS-MDP as follows.

Definition 2.2. *Contextual Situational MDP (CS-MDP)* is a tuple $(\mathcal{Z}, \mathcal{S}, \mathcal{A}, \mathcal{C}, p_s, r, \gamma, \mathcal{M}(z))$, where $p_s(s_{t+1} \mid s_t, a_t, c_t)$ is the state transition, $r(s_t, a_t, c_t)$ is the reward

function. $\mathcal{Z}$ is called super-context space, and $\mathcal{M}$ is a function mapping any super-context $z \in \mathcal{Z}$ to a situational MDP $\mathcal{M}(z) = (\mathcal{S}, \mathcal{A}, \mathcal{C}, p_z, p_s, r, \gamma)$. We use $p_z(c_{t+1} \mid c_t)$ to denote the situation dynamic, and emphasize its dependency upon super-context z.

The context z is latent, while the states s and situations c are observable. For any $z_1 \neq z_2$, their corresponding situational MDPs, $\mathcal{M}(z_1)$ and $\mathcal{M}(z_2)$, share the same state transition and reward function but differ only in their situation dynamic, p_{z_1} and p_{z_2}. CS-MDP arises naturally in a variety of real-world problems. Taking inventory control as an example, c refers to the fluctuated user demand, while z refers to the demand pattern which may vary by products and cities. Demand for seasonal products like ice cream may peak in summer, while demand for daily necessities may fluctuate slightly. Other instances of z may include the outdoor temperature changes pattern in the field of Heating, Ventilation and Air Conditioning (HVAC), the economic cycle in stock trading, the task arriving patterns in queuing systems, etc. To avoid confusion, from now on, we refer to c_t in Situational MDP as *situation*, and z in Contextual Situational MDP as *context* or *super-context*.

Generalization to unseen test environments Motivated by real world applications, we highlight the problem of generalization under CS-MDP in this paragraph. Instead of training and testing on exactly the same environment, we aim to address a more realistic setting where test environments may be different from those used in training, which is quite general since it is almost impossible to train our algorithms in all possible contexts. Intuitively, as shown in Figure 1, training environments and test environments may have context sampled from different distributions. To be specific, given a set of training environments $\{\mathcal{M}(z^i_{train})\}$ with context z^i_{train} sampled from certain distribution $p_{train}(z)$, we aim to train an agent that can adapt to a set of test environments $\{\mathcal{M}(z^i_{test})\}$ with possibly unseen contexts z^i_{test} sampled from $p_{test}(z)$ without further online training. The training environments $\{\mathcal{M}(z^i_{train})\}$ and test environments $\{\mathcal{M}(z^i_{test})\}$ are under the same CS-MDP, which means they share the same state transition and reward function but differ only in their situation dynamic, $p_{z_{train}}(c_{t+1}|c_t)$ and $p_{z_{test}}(c_{t+1}|c_t)$.

Offline situation data for test environments We further assume we have access to offline situation data $\{c_t\}^T_{t=0}$ for the test environment. Since the situation dynamics are independent of agents' actions, it is usually possible to collect offline situation data without interacting with the environments as shown in Figure 1. For instance, actually deploying an agent might be expensive. Still, we can easily collect the history of customers' orders for some specific products, the outdoor temperature in some distant cities, and the price of stocks that are not actually held at a small cost. By exploiting the pattern inside the collected data, we are exempt from re-training our agent in an expensive online manner. The agent only needs to learn to adapt given the corresponding offline situation data under the same CS-MDP.

3. Methodology

In this section, we present our algorithm which addresses the problem of adapting to unseen situational MDPs with offline situation data. Our method consists of two components:

- *Context-aware generalization* that learns a context-dependent policy based on the inferred belief distribution over the super-context.
- *Off-dynamic learning* that enables policy optimization for the test Situational MDP without further online training.

Context-aware generalization enables the agent to adapt to test environments by optimizing policy behavior on training environments with similar situation dynamics, where the similarity is encoded in the latent super-context z . In Section 3.1, we introduce the inference of the latent super-context, and its belief is incorporated into the input for the policy, to allow the agent to incorporate the inferred context and context uncertainty into its decision-making and generalize to similar unseen test environments.

Off-dynamic learning optimizes the agent's policy on test environments directly, with offline situation data for the test environment and online interactions from training environments. In Section 3.2, we extend the algorithm to allow for off-dynamic learning, for which we present convergence results under the condition that training environments are informative enough with respect to test environments.

3.1. Context-aware Generalization

In this part, we focus on improving testing performance through context-aware generalization without direct interactions with test environments.

To be specific, we learn a context-dependent policy $\pi(\cdot|s_t, c_t, z)$ conditions on the inferred context z and optimize the expected returns on training environments through online interaction:

$$\mathbb{E}_{z\sim p_{train}(z)}\left[\mathbb{E}_{p_s,p_z,\pi}\Big[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t, c_t)\Big]\right] \quad (1)$$

Though we do not directly train our policy on test environments, we can still improve the testing performance

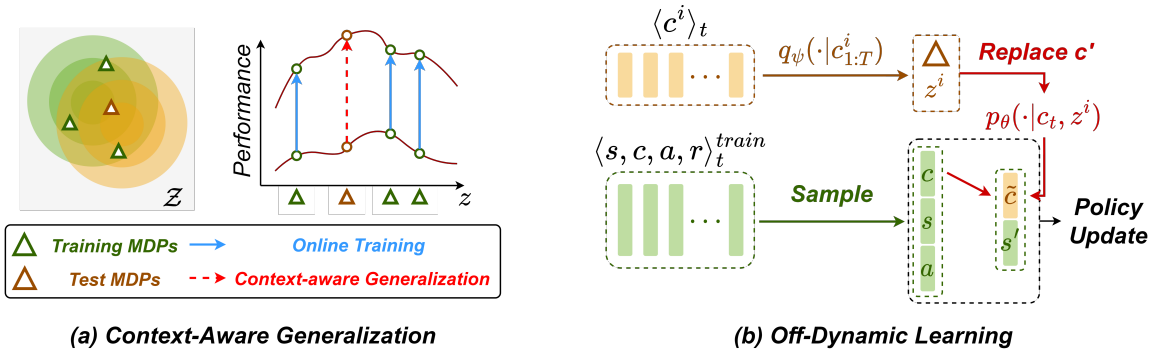

Figure 2. An overview of our method. *(a) Context-aware generalization*. During training, our agent optimizes the policy on training environments (denoted in green triangles) through online interactions (denoted in blue arrow). By extracting the situation pattern z, COGOL can improve testing performance (denoted in the orange triangle) through context-aware generalization (denoted in red dashed arrow) without direct interactions. *(b) Off-dynamic learning*. We use offline situation data $\langle c^i \rangle_t$ to extract the context z^i, and replace the sampled transition tuple by $(s, c, a, s', \tilde{c})$, where $\tilde{c} \sim p_\theta(\cdot|c, z^i)$.

through generalization. This is based on an observation that environments with similar situation dynamics are close in nature and therefore may require a similar policy. For instance, demands for ice cream and cold drinks may experience almost identical seasonal fluctuations and therefore may require the same inventory management policy. Therefore, we can exploit this observation by conditioning our policy on the underlying z to make the agent aware of the context and improve the performance on similar environments.

Context Inference It is crucial to capture the underlying situation dynamics pattern accurately, namely infer the latent context z to achieve context-aware generalization. In COGOL, We propose to infer z using the variational inference framework. To be specific, let q_ϕ denote the posterior inference network parameterized by ϕ and p_θ denote the decoder parameterized by θ. Then, at time step t, the variational lower bound for the log-likelihood of the situation sequence is given by

$$\log p(c_{1:T}) \geq \mathop{\mathbb{E}}_{q_\phi} \left[\log p_\theta(c_{1:T}|z)\right] - \mathbf{D}_{KL}\Big(q_\phi(z|c_{1:t}) \| p(z)\Big)$$

where $p(z)$ refers to the prior distribution of context. The derivation can be found in Appendix A. The reconstruction term can be factorized as

$$\log p_\theta(c_{1:T}|z) = \sum_{t=1}^{T} \log p_\theta(c_t|c_{t-1}, z)$$

which decodes the whole trajectory using inferred $z \sim q_\phi(z|c_{1:t})$. By definition, the context z remains the same throughout the episode. Therefore to make the agent aware of the context as early in the episode as possible, we decode the whole trajectory using the inferred at time step t: $z \sim q_\phi(z|c_{1:t})$. To ease the computation burden, we use a randomly sampled reconstruction term by randomly choosing a time step $\tau \sim \mathcal{U}[1, T]$ and replacing the reconstruction term with a single step reconstruction for time step τ. The loss function can therefore be denoted as

$$\mathcal{J}_M = \mathop{\mathbb{E}}_{q_\phi}[\log p_\theta(c_\tau|c_{\tau-1}, z)] - \mathbf{D}_{KL}\Big(q_\phi(z|c_{1:t}) \| p(z)\Big) \tag{2}$$

Policy optimization with context belief At test time, we use the learned encoder $q_\phi(\cdot)$ to infer the current context $q_\phi(z|c_{1:t})$. Since the context z is vague, it is desirable to take a more conservative policy when the inference result is highly uncertain and vice versa. To incorporate the uncertainty into decision making, instead of learning a policy directly conditions on z, we propose to augment the state as $(s, c, b) \in \mathcal{S} \times \mathcal{C} \times \mathcal{B}$, where $\mathcal{B}$ is the context belief space, as inspired by Kaelbling et al. (1998). This can help the agent to aware not only of the context but also of the uncertainty of the context. The augmentation implicitly defines an augmented MDP with reward function $R(s_t, a_t, b_t)$ state transition

$$\begin{aligned} &p_b(s_{t+1}, c_{t+1}, b_{t+1}|s_t, c_t, a_t, b_t) \\ &= p(s_{t+1}|s_t, a_t, c_t)p(c_{t+1}|c_t, b_t)p(b_{t+1}|b_t, c_{t+1}) \end{aligned}$$

The reward function $R(s_t, a_t, b_t)$ and context transition $p(c_{t+1}|c_t, b_t)$ are taking expectations over $b_t(z)$, and the belief transition $p(b_{t+1}|b_t, c_{t+1})$ is specified by the learned encoder q_ϕ. As more situations c are observed, the uncertainty in the inferred belief distribution b will gradually decrease and the agent will be more informed and can learn

to behave more actively by maximizing

$$\mathcal{J}_\pi = \mathbb{E}_{z\sim p_{train}(z)}\left[\mathbb{E}_{p_s,p_z,\pi}\Big[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t,b_t)\Big]\right] \quad (3)$$

3.2. Off-dynamic Learning

In this section, we extend our algorithm to allow for directly optimizing testing performance by off-dynamic learning, using only offline situation data and without further online training.

To formulate, suppose we have access to offline situation data $\{c^i_{1:T}\}_i$ for test environments $z^i_{test}, i \in \{1, 2, \cdots, K\}$. Off-dynamic learning enables the agent to directly improve expected returns taken over $p_{test}(z)$:

$$\mathbb{E}_{z\sim p_{test}(z)}\left[\mathbb{E}_{p_s,p_z,\pi}\Big[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t,c_t)\Big]\right]$$

Instead of interacting on these test environments, off-dynamic improves test performance using interactions collected on training environments with context $\{z^i_{train}\}$.

To be specific, we first train our encoder q_ϕ and decoder p_θ by the objective in Equation (2) further using the offline situation data. Then, during online interactions with training environments, we propose to augment the training data using the situations given by the decoder. Denote a transition tuple collected during online interaction as $(s_t, c_t, z_t, a_t, r_t, s_{t+1}, c_{t+1}, z_{t+1})$, where $z_t \sim q_\phi(z|c_{1:t})$. Instead of directly using this tuple to train our policy as in Section 3.1, we propose to replace the z_t and c_{t+1} according to the test environments to perform off-dynamic learning.

The off-dynamic learning procedure is described as follows: We randomly select one trajectory $c^i_{1:T}$ from the offline dataset and extract the context $z^i \sim q_\phi(z|c^i_{1:T})$. Then, we replace z_t, z_{t+1} in the tuple by z^i and replace c_{t+1} via $\tilde{c}$ resampled from the decoder $\tilde{c} \sim p_\theta(c|c_t, z^i)$. Since $p(s_{t+1}|s_t, c_t, a_t)$ and $r_t(s_t, c_t, a_t)$ are not affected by this replacement, it is safe to use the augmented tuple $(s_t, c_t, z^i, a_t, r_t, s_{t+1}, \tilde{c}, z^i)$ to perform policy updates.

However, there is still a lingering question to be answered: *how large the derivation between training and test environments is allowed*? Consider an extreme corner case where situations $\{c^i_{1:T}\}_i$ observed during training are completely disjoint with situations $\{c^i_{1:T}\}_i$ observed during testing, then the learned encoder might be erroneous when trying to reconstruct $p_\theta(c|c_t, z^i)$. We give the following theorem:

Theorem 3.1 (Convergence condition of off-dynamic learning)**.** *Given a test situational MDP $M(z_0)$ with context z_0, and a set of training situational MDPs $\{M(z^i_{train})\}_i$, under finite $\mathcal{S}, \mathcal{C}, \mathcal{A}$, the learned Q function can converge to its optimal Q function $Q^*_{z_p}$ if, for all $c, c' \in \mathcal{C}_{z_0}$ with $p_{z_0}(c'|c) > 0$, we have*

$$\sum_i p_{z^i_{train}}(c'|c) > 0$$

The full statement and the proof can be found in Appendix B. Intuitively, as long as the transition tuple (c, c') of the target test environment has shown up in any of the training environments, the value learning will converge. This is in contrast with Section 3.1, where the context-aware generalization requires the training environments to have a similar pattern to generalize to testing environments. Off-dynamic learning only requires the context transition tuples that have shown up in the training process, regardless of how close z^i_{train} is to z_{test}.

3.3. Algorithm and Implementations of COGOL

Algorithm 1 COGOL Algorithm

Input: training environments $\mathcal{M}\left(z^i_{train}\right)$, offline situation dataset $\{c^i_{1:T}\}_i$, variational encoder q_ϕ and decoder p_θ, policy π_ψ and value function Q_ψ, replay buffer $\mathcal{D}_{RL}$, training situation buffer $\mathcal{D}_s$
Train q_ϕ, p_θ on offline situation sequences $\{c^i_{1:T}\}_i$ using Equation (2).
for each iteration **do**
  for each environment step **do**
    Infer context belief $b_t = q_\phi(z|c_{1:t})$ using history contexts
    Select action $a_t \sim \pi_\psi(s_t, c_t, b_t)$
    Update context belief $b_{t+1} = q_\psi(z \mid c_{1:t+1})$
    Add $(s_t, c_t, b_t, a_t, r_t, s_{t+1}, c_{t+1}, b_{t+1})$ to $\mathcal{D}_{RL}$
    Add $(s_t, c_t, \tilde{b}, a_t, r_t, s_{t+1}, \tilde{c}, \tilde{b})$ to $\mathcal{D}_{RL}$
    if *done* **then**
      Add observed situation sequence $c_{1:T}$ to $\mathcal{D}_s$
    end if
  end for
  for each gradient step **do**
    Update π_ψ and Q_ψ using samples from $\mathcal{D}_{RL}$ according to Equation (4)
    Update q_ϕ and p_θ using samples from $\mathcal{D}_s$ according to Equation (2)
  end for
end for

In this section, we describe the overall algorithm and implementation details of COGOL. To align with Section 3.1, we extend the policy described in Section 3.2 by augmenting the state with the belief state. We build our RL algorithm based on SAC (Haarnoja et al., 2018) to learn the policy $\pi_\psi(s_t, c_t, b_t)$ as well as the $Q_\psi(s_t, c_t, b_t)$. During the online interaction as described in Section 3.2, we store the collected transition tuples as well as the replaced transition tuples into the replay buffer $\mathcal{D}_{RL}$, so as to maximize

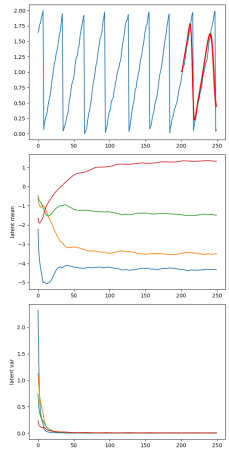

Figure 3. Posterior belief is updated as more situation is observed. In the top figure we use blue curve to present the situation sequence. The red curve is the output of decoder after observed situation sequence for 200 steps, which matches the original sequence well. In middle and buttom we show how latent mean and latent variance evolves as more situation transitions are observed.

the expected total return

$$\mathcal{J}_{RL} = \mathbb{E}_z \left[\mathbb{E}_{p_s, p_z, \pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, b_t) \right] \right] \tag{4}$$

where z takes expectation over a mixed distribution of $p_{train}(z)$ and $p_{test}(z)$. We summarize the whole framework in Algorithm 1.

4. Experiments

We start our experiments with a simple 1D object tracking task and then generalize onto large-scale MuJoCo environments.

4.1. 1D Object Tracking

In this experiment, we test our algorithm on a 1-D object tracking environment. The environment includes a robot moving along a one-dimensional line, and the goal of this robot is to chase a moving target moving along the same line. The position of the target serves as the situation process and is controlled by a scalar z. The speed and acceleration capacities of the robot is limited. Therefore, , it must learn to capture the situation dynamics c in advance so as to infer the future positions of the target in order to maximize the reward.

First, we show in Figure 3 that our algorithm is able to maintain a posterior belief distribution over context process dynamics as more situations c are observed. The posterior variance is decreased towards zero, while posterior mean converges to a fixed point. Next, we show in Figure 4 that

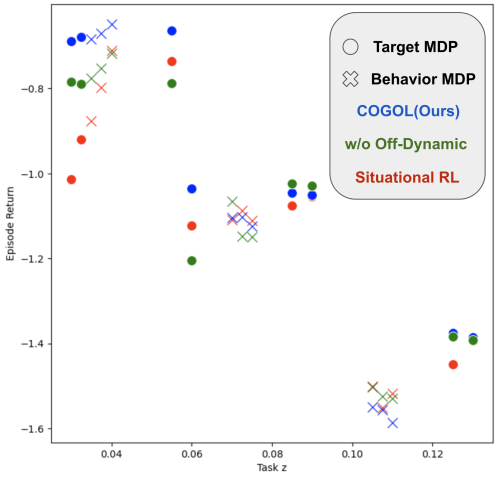

Figure 4. The experiment results on 1-D object tracking task.

when there are gaps in the training tasks or the target tasks show a distribution shift from the training tasks, our algorithm is still able to derive a trust-worthy policy without interactions with the corresponding environment directly. In this figure, 'x' indicates tasks for training, whereas 'o' indicates tasks for testing, e.g. a red 'o' refers to the average episode return of Situational RL on a sampled test MDP. Note that the benefits of context-aware generalization and off-dynamic learning are more substantial on test MDPs that deviates more from training MDPs.

4.2. Locomotion Control Tasks with Varied Situation Dynamics

We show that COGOL can achieve superior testing performance on unseen test environments. We conduct our experiments on a modified MuJoCo (Todorov et al., 2012) tasks - Half Cheetah Goal. In this task, the policy needs to move towards a moving goal to obtain positive rewards. Situation dynamics for both training and testing environments have a piece-wise constant structure, but training environments have a decreasing pattern, and test environments have an increasing pattern (Figure 6). We evaluate the agent's performance on both training and test environments, and compare our algorithm with Situational RL. The results are provided in Figure 5. It can be seen from Figure 5 that, COGOL achieves significantly better performance on test environments, and slightly improves on training environments.

5. Related Works

Situational MDP Situational MDP (Chen et al., 2022), which is also known as Input-driven MDP (Mao et al., 2018b) or MDP with exogenous processes (Chitnis & Lozano-Pérez, 2020; Boutilier et al., 1999), solves a series of problems with dynamic environmental factors (i.e., sit-

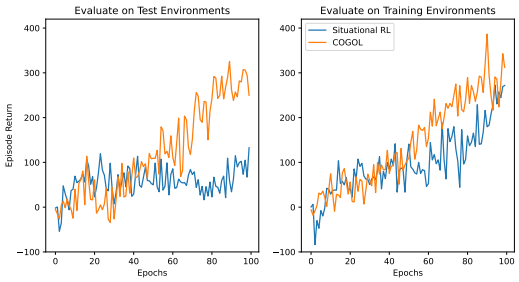

Figure 5. The experiment results on Half Cheetah Goal

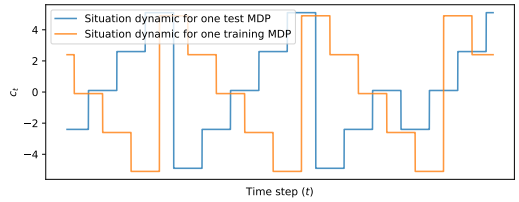

Figure 6. Situation dynamic used in Half Cheetah Goal. For both training and test environments, we randomly sample one super-context z, and generate one situation process path with the sampled z.

uations). Notably, the factors can significantly affect state transitions and reward functions but are beyond the control of the agent. Existing works try to solve challenges in different aspects of situational MDPs.

A branch of works (Dietterich et al., 2018; Efroni et al., 2021; Chitnis & Lozano-Pérez, 2020; Wang et al., 2022b; Yang et al., 2022; Wang et al., 2022a) focuses on the setting where the observations is a mixture of endogenous states and exogenous situations. Different approaches have been proposed to separate states and situations from observations for variance reduction (Yang et al., 2022), compact world model learning (Chitnis & Lozano-Pérez, 2020; Wang et al., 2022a), or efficient dynamics learning (Wang et al., 2022b).

Policy learning under situational MDPs also receives attention from researchers. Mao et al. (2018b) proposes to reduce the variance in policy gradient methods by incorporating the input situation into the baselines. Yang et al. (2022) conditions the policy on the future outcomes to improve performance on highly stochastic environments.

Another stream of works such as (Ren et al., 2021; Feng et al., 2022; Kumar et al., 2021; Chen et al., 2022) operates in the setting where the situations are unobservable, where the models learn to infer the latent situations. Ren et al. (2021); Feng et al. (2022) assumes a Markovian situation transition, while Chen et al. (2022) assumes a piecewise stable situation evolution pattern and learns to detect the abrupt changes of the latent situations.

However, these methods only focus on solving a single instance of situational MDP. In our paper, we aim to solve a set of situational MDPs with different situation dynamics at the same time and enable our agent to adapt to possibly unseen situation dynamics at test time without further online training. Existing works on Situational MDPs lack the relevant mechanisms to enable efficient generalization.

Contextual MDP Contextual MDP (Hallak et al., 2015; Modi et al., 2018) can be formulated as a form of generalization. They assume there exists a latent context that affects the state transition and reward function which may vary from episode to episode. For instance, works including (Perez et al., 2020; Xie et al., 2022; Ajay et al., 2022) propose to generalize to novel and unseen tasks. Another branch of work focuses on task transferring in Contextual MDPs (Klink et al., 2020; Eimer et al., 2021; Hu et al., 2018; Tirinzoni et al., 2020). Methods belonging to this branch include designing curriculum learning (Klink et al., 2020; Eimer et al., 2021), disentangling tasks and environments (Hu et al., 2018), or through a learned generative model (Tirinzoni et al., 2020). Other works in Meta RL like (Finn et al., 2017; Zintgraf et al., 2019; Al-Shedivat et al., 2018; Vuorio et al., 2019; Humplik et al., 2019; Poiani et al., 2021) also learn to adapt to new tasks efficiently. However, these lines of work do not treat the endogenous state and situations separately and the contexts need to be inferred from the interactions. In many real world applications, the differences in tasks only exist in situation dynamics, which makes inferring from full trajectories redundant and vulnerable to spurious correlations (Wang et al., 2022b). This also hinders the agent from using the offline situation dataset which contains only the context trajectories without states and actions from some behavior policy.

6. Conclusion

Real-world applications, such as inventory control, present new challenges to RL, including the dynamic uncontrollable environmental factors, and the need for generalization between environments which differ only in such situation dynamics. In this paper, we address the above challenge by formulating the problem as Contextual Situational MDP, and propose COGOL, a deep RL algorithm that can adapt to new situation dynamics without further online training, by ***CO**ntext-aware **G**eneralization* and ***O**ff-dynamics **L**earning* (COGOL). Experiment results demonstrate that our trained agent can effectively adapt to new situation dynamics. For future work, we plan to study generalization bounds from a theoretical perspective for Contextual Situational MDP, which presents a more interesting problem than the generalization bounds in Contextual MDP, with

additional offline situation data on test environments.

References

Ajay, A., Gupta, A., Ghosh, D., Levine, S., and Agrawal, P. Distributionally adaptive meta reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

Al-Shedivat, M., Bansal, T., Burda, Y., Sutskever, I., Mordatch, I., and Abbeel, P. Continuous adaptation via meta-learning in nonstationary and competitive environments. In *International Conference on Learning Representations*, 2018.

Boutilier, C., Dean, T., and Hanks, S. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11: 1–94, 1999.

Chen, X., Zhu, X., Zheng, Y., Zhang, P., Zhao, L., Cheng, W., CHENG, P., Xiong, Y., Qin, T., Chen, J., and Liu, T.-Y. An adaptive deep RL method for non-stationary environments with piecewise stable context. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=5swt6zUFrVp.

Chitnis, R. and Lozano-Pérez, T. Learning compact models for planning with exogenous processes. In *Conference on Robot Learning*, pp. 813–822. PMLR, 2020.

Dietterich, T. G., Trimponias, G., and Chen, Z. Discovering and removing exogenous state variables and rewards for reinforcement learning. *ArXiv*, abs/1806.01584, 2018.

Efroni, Y., Misra, D. K., Krishnamurthy, A., Agarwal, A., and Langford, J. Provable rl with exogenous distractors via multistep inverse dynamics. *ArXiv*, abs/2110.08847, 2021.

Eimer, T., Biedenkapp, A., Hutter, F., and Lindauer, M. Self-paced context evaluation for contextual reinforcement learning. In *International Conference on Machine Learning*, pp. 2948–2958. PMLR, 2021.

Feng, F., Huang, B., Zhang, K., and Magliacane, S. Factored adaptation for non-stationary reinforcement learning. *arXiv preprint arXiv:2203.16582*, 2022.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

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.

Hallak, A., Di Castro, D., and Mannor, S. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.

Hu, H., Chen, L., Gong, B., and Sha, F. Synthesized policies for transfer and adaptation across tasks and environments. *Advances in Neural Information Processing Systems*, 31, 2018.

Humplik, J., Galashov, A., Hasenclever, L., Ortega, P. A., Teh, Y. W., and Heess, N. Meta reinforcement learning as task inference. *arXiv preprint arXiv:1905.06424*, 2019.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Klink, P., Abdulsamad, H., Belousov, B., and Peters, J. Self-paced contextual reinforcement learning. In *Conference on Robot Learning*, pp. 513–529. PMLR, 2020.

Kumar, A., Fu, Z., Pathak, D., and Malik, J. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

Mao, H., Venkatakrishnan, S. B., Schwarzkopf, M., and Alizadeh, M. Variance reduction for reinforcement learning in input-driven environments. *ArXiv*, abs/1807.02264, 2018a.

Mao, H., Venkatakrishnan, S. B., Schwarzkopf, M., and Alizadeh, M. Variance reduction for reinforcement learning in input-driven environments. *arXiv preprint arXiv:1807.02264*, 2018b.

Modi, A., Jiang, N., Singh, S., and Tewari, A. Markov decision processes with continuous side information. In *Algorithmic Learning Theory*, pp. 597–618. PMLR, 2018.

Pan, M., Zhu, X., Wang, Y., and Yang, X. Iso-dream: Isolating noncontrollable visual dynamics in world models. volume abs/2205.13817, 2022.

Perez, C., Petroski Such, F., and Karaletsos, T. Generalized hidden parameter mdps:transferable model-based rl in a handful of trials. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5403–5411, Apr. 2020. doi: 10.1609/aaai.v34i04.5989.

Poiani, R., Tirinzoni, A., and Restelli, M. Meta-reinforcement learning by tracking task non-stationarity. *arXiv preprint arXiv:2105.08834*, 2021.

Ren, H., Sootla, A., Jafferjee, T., Shen, J., Wang, J., and Ammar, H. B. Reinforcement learning in presence of discrete markovian context evolution. In *International Conference on Learning Representations*, 2021.

Tirinzoni, A., Poiani, R., and Restelli, M. Sequential transfer in reinforcement learning with a generative model. In *International Conference on Machine Learning*, pp. 9481–9492. PMLR, 2020.

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.

Van Hasselt, H. P. et al. *Insights in reinforcement rearning: formal analysis and empirical evaluation of temporal-difference learning algorithms*. Utrecht University, 2011.

Vuorio, R., Sun, S.-H., Hu, H., and Lim, J. J. Multimodal model-agnostic meta-learning via task-aware modulation. *Advances in Neural Information Processing Systems*, 32, 2019.

Wang, T., Du, S., Torralba, A., Isola, P., Zhang, A., and Tian, Y. Denoised mdps: Learning world models better than the world itself. In *International Conference on Machine Learning*, pp. 22591–22612. PMLR, 2022a.

Wang, Z., Xiao, X., Xu, Z., Zhu, Y., and Stone, P. Causal dynamics learning for task-independent state abstraction. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23151–23180. PMLR, 17–23 Jul 2022b.

Xie, A., Sodhani, S., Finn, C., Pineau, J., and Zhang, A. Robust policy learning over multiple uncertainty sets. *arXiv preprint arXiv:2202.07013*, 2022.

Yang, M., Schuurmans, D., Abbeel, P., and Nachum, O. Dichotomy of control: Separating what you can control from what you cannot. *arXiv preprint arXiv:2210.13435*, 2022.

Zintgraf, L., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., and Whiteson, S. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*, 2019.

A. Derivation of ELBO

The ELBO in Section 3.1 can be derived as follows:

$$\begin{aligned}
&\log p(c_{1:T}) = \log \mathbb{E}_{q_\phi}\left[\frac{p(z)}{q_\phi(z|c_{1:t})} p_\theta(c_{1:T}|z)\right] \\
&\geq \mathbb{E}_{q_\phi}\left[\log p_\theta(c_{1:T}|z) + \log \frac{p(z)}{q_\phi(z|c_{1:t})}\right] \\
&= \mathbb{E}_{q_\phi}\left[\log p_\theta(c_{1:T}|z)\right] - \mathbf{D}_{KL}\left(q_\phi(z|c_{1:t})\|p(z)\right)
\end{aligned}$$

B. Off-dynamic Learning Convergence Guarantee

We restate our theorem as below by first define the off-context learning updates Q-values using the update rule

$$\begin{aligned}
Q_{t+1}(s_t, a_t, c_t) \leftarrow (1 - \alpha_t(s_t, a_t, c_t))Q_t(s_t, a_t, c_t) \\
+\alpha_t(s_t, a_t, c_t)y(r_t, c_t, s_{t+1}) \\
Q_{t+1}(s, a, c) \leftarrow Q_t(s, a, c), \forall (s, a, c) \neq (s_t, a_t, c_t)
\end{aligned} \tag{5}$$

where $\alpha_t(s_t, a_t, c_t)$ is learning rate, with

$$\begin{aligned}
&y(r_t, c_t, s_{t+1}) \\
:=&r_{t+1} + \gamma \sum_{c'} p_{z_0}(c' \mid c_t) \sum_{a'} \pi_t(a' \mid s_{t+1}, c') Q(s_{t+1}, a', c').
\end{aligned}$$

Then, the convergence condition of can be restated as follows:

Theorem B.1 (Convergence condition of off-dynamic learning)**.** *Given a test situational MDP $M(z_0)$ with context z_0, and a set of training situational MDPs $\{M(z^i_{train})\}_i$. With off-dynamic learning as defined by Equation (5), Q_t converges to its optimal Q-function $Q^*_{z_0}$ whenever the following assumptions hold:*

1. *$\mathcal{S}, \mathcal{A}$ and $\mathcal{C}$ are finite.*
2. *$\alpha_t(s_t, a_t, c_t) \in [0, 1]$, $\sum_t \alpha_t(s_t, a_t, c_t) = \infty$, $\sum_t (\alpha_t(s_t, a_t, c_t))^2 < \infty$ w.p.1 and $\forall (s, a, c) \neq (s_t, a_t, c_t)$: $\alpha_t(s, a, c) = 0$.*
3. *The policy π_t is greedy in the limit and the behavior policy ensures infinite exploration*
4. *for all $c, c' \in \mathcal{C}_{z_0}$ with $p_{z_0}(c'|c) > 0$, we have $\sum_i p_{z^i_{train}}(c'|c) > 0$.*
5. *$\mathrm{Var}[\mathcal{R}(s, a, c)] < \infty, \forall (s, a, c) \in \mathcal{S} \times \mathcal{A} \times \mathcal{C}$.*

The proof of Theorem B.1 is based on Lemma B.2 (Van Hasselt et al., 2011).

Lemma B.2. *Consider a stochastic process (ζ_t, Δ_t, F_t), $t \geq 0$, where $\zeta_t, \Delta_t, F_t : X \to \mathbb{R}$, which satisfies the equations*

$$\Delta_{t+1}(x) = (1 - \zeta_t(x))\Delta_t(x) + \zeta_t(x) F_t(x), \forall x \in X, t = 0, 1, 2, \ldots.$$

Let P_t be a sequence of increasing σ-fields such that ζ_0 and Δ_0 are P_0-measurable and ζ_t, Δ_t and F_{t-1} are P_t-measurable, $t \geq 1$. Assume that the following hold:

1. *the set X is finite,*
2. *$\zeta_t(x) \in [0, 1]$, $\sum_t \zeta_t(x) = \infty$, $\sum_t \zeta^2_t(x) < \infty$ w.p.1 and $\forall x \neq x_t : \zeta_t(x) = 0$,*
3. *$\|\mathbb{E}[F_t \mid P_t]\| \leq \kappa \|\Delta_t\| + c_t$, where $\kappa \in [0, 1)$ and c_t converges to zero w.p.1,*
4. *$\mathrm{Var}[F_t(x) \mid P_t] \leq K(1 + \kappa \|\Delta_t\|)^2, \forall x \in X$, where K is some constant,*

where $\|\cdot\|$ denotes a maximum norm[2]*. Then Δ_t converges to zero with probability one.*

The idea to prove Theorem B.1 is to apply Lemma B.2 with $X = \mathcal{S} \times \mathcal{A} \times \mathcal{C}$, $x_t = (s_t, a_t, c_t)$, $\zeta_t(x_t) = \alpha_t(s_t, a_t, c_t)$ and $\Delta_t(x_t) = Q_t(s_t, a_t, c_t) - Q^*(s_t, a_t, c_t)$. If we can prove that Δ_t converges to zero with probability one, we have convergence of the Q-values to the optimal values.

Proof. Denote the σ-field generated by the random variables $\{s_t, \alpha_t, a_t, c_t, r_{t-1}, \ldots, s_1, \alpha_1, a_1, c_1, Q_0\}$ as P_t. Note that $Q_t, Q_{t-1}, \ldots, Q_0$ are P_t-measurable and, thus, both Δ_t and F_{t-1} are P_t-measurable, satisfying the measurability conditions of Lemma B.2.

To prove this theorem, we simply check that all the conditions of Lemma B.2 are satisfied. The first condition of this lemma is ensured by the first assumption of the theorem. The second condition holds by definition of the learning rate α. The satisfaction of the third condition of Lemma B.2 is given below.

Given P_t, for $(s, a, c) \neq (s_t, a_t, c_t)$, we can define $F_t(s, a, c)$ as zero. We can derive $F_t(s_t, a_t, c_t)$ as

$$\begin{aligned} F_t(s_t, a_t, c_t) &= \frac{\Delta_{t+1}(x_t) - (1 - \zeta_t(x))\,\Delta_t(x_t)}{\zeta_t(x_t)} \\ &= \mathcal{R}(s_t, a_t, c_t) + \gamma \sum_{c'} P^e(c_t, c') \sum_{a'} \pi_t^e(s_{t+1}, c', a')\, Q_t(s_{t+1}, a', c') - Q^*(s_t, a_t, c_t) \end{aligned}$$

given $x_t = (s_t, a_t, c_t)$, $\zeta_t(x_t) = \alpha_t(s_t, a_t, c_t)$, $\Delta_t(x_t) = Q_t(s_t, a_t, c_t) - Q^*(s_t, a_t, c_t)$ and the off-dynamic update rule (5). Thus $\mathbb{E}[F_t(s, a, c) \mid P_t] = 0, \forall (s, a, c) \neq (s_t, a_t, c_t)$, and

$$\begin{aligned} &\mathbb{E}[F_t(s_t, a_t, c_t) \mid P_t] \\ =&\mathbb{E}_{s_{t+1}, r_t(s_t, a_t, c_t)}[\mathcal{R}(s_t, a_t, c_t) + \gamma \sum_{c'} P^e(c_t, c') \sum_{a'} \pi_t^e(s_{t+1}, c', a')\, Q_t(s_{t+1}, a', c') - Q^*(s_t, a_t, c_t)] \\ =&\mathbb{E}_{s', r_t(s,a,c)}[r_t(s, a, c) + \gamma \sum_{c'} P^e(c, c') \sum_{a'} \pi_t^e(s', a', c')\, Q_t(s', a', c') - Q^*(s, a, c)] \\ =&\mathbb{E}_{s', r_t(s,a,c)}[r_t(s, a, c) + \gamma \sum_{c'} P^e(c, c') \sum_{a'} \pi_t^e(s', a', c')\, Q_t(s', a', c')] - Q^*(s, a, c) \\ =&R_{sac} + \gamma \sum_{s'} P_{sac}^{s'} \left(\sum_{c'} P^e(c, c') \sum_{a'} \pi_t^e(s', c', a') Q_t(s', a', c') \right) - Q^*(s, a) \\ =&R_{sac} + \gamma \sum_{s'} P_{sac}^{s'} \left(\sum_{c'} P^e(c, c') \sum_{a'} \pi_t^e(s', c', a') Q_t(s', a', c') \right) - \\ &\left(\sum_{c'} P^e(c, c') \sum_{s'} P_{sac}^{s'} \left(\gamma \max_{a'} Q^*(s', a', c') \right) + R_{sac} \right) \\ =&\gamma \sum_{s'} P_{sac}^{s'} \left(\sum_{c'} P^e(c, c') \sum_{a'} \pi_t^e(s', c', a') Q_t(s', a', c') - \sum_{c'} P^e(c, c') \max_{a'} Q^*(s', a', c') \right). \end{aligned}$$

To verify the satisfaction of the third condition of Lemma B.2,

$$\begin{aligned} &\|\mathbb{E}[F_t \mid P_t]\|_\infty \\ &= \max_{s,a,c} |\mathbb{E}[F_t(s, a, c) \mid P_t]| \\ &\leq |\mathbb{E}[F_t(s_t, a_t, c_t) \mid P_t]| \quad \text{since } \mathbb{E}[F_t(s, a, c) \mid P_t] = 0 \text{ for } (s, a, c) \neq (s_t, a_t, c_t) \\ &= \gamma \left| \sum_{s'} P_{s_t a_t c_t}^{s'} \left(\sum_{c'} P^e(c_t, c') \sum_{a'} \pi_t^e(s', c', a') Q_t(s', a', c') - \sum_{c'} P^e(c_t, c') \max_{a'} Q^*(s', a', c') \right) \right| \end{aligned}$$

[2] $\|\mathbb{E}[F_t \mid P_t]\| = \max_{x \in X} |\mathbb{E}[F_t(x) \mid P_t]|$, and $\|\Delta_t\| = \max_{x \in X} |\Delta_t(x)|$.

$$
\begin{aligned}
&= \gamma \left| \sum_{s'} P^{s'}_{s_t a_t c_t} \sum_{c'} P^e(c_t, c') \left(\sum_{a'} \pi^e_t(s', c', a') Q_t(s', a', c') - \max_{a'} Q^*(s', a', c') \right) \right| \\
&\leq \gamma \sum_{s'} P^{s'}_{s_t a_t c_t} \sum_{c'} P^e(c_t, c') \left| \sum_{a'} \pi^e_t(s', c', a') Q_t(s', a', c') - \max_{a'} Q^*(s', a', c') \right| \\
&\leq \gamma \sum_{s'} P^{s'}_{s_t a_t c_t} \sum_{c'} P^e(c_t, c') \left(\left| \max_{a'} Q_t(s', a', c') - \max_{a'} Q^*(s', a', c') \right| + \right. \\
&\qquad \left. \left| \sum_{a'} \pi^e_t(s', c', a') Q_t(s', a', c') - \max_{a'} Q_t(s', a', c') \right| \right) \\
&\leq \gamma \max_{s'} \max_{c'} \left(\left| \max_{a'} Q_t(s', a', c') - \max_{a'} Q^*(s', a', c') \right| + \left| \sum_{a'} \pi^e_t(s', c', a') Q_t(s', a', c') - \max_{a'} Q_t(s', a', c') \right| \right) \\
&= \gamma \max_{s'} \max_{c'} \left| \max_{a'} Q_t(s', a', c') - \max_{a'} Q^*(s', a', c') \right| + \\
&\qquad \gamma \max_{s'} \max_{c'} \left| \sum_{a'} \pi^e_t(s', c', a') Q_t(s', a', c') - \max_{a'} Q_t(s', a', c') \right| \\
&\leq \gamma \max_{s'} \max_{c'} \max_{a'} |Q_t(s', a', c') - Q^*(s', a', c')| + \gamma \max_{s'} \max_{c'} \left| \sum_{a'} \pi^e_t(s', c', a') Q_t(s', a', c') - \max_{a'} Q_t(s', a', c') \right|
\end{aligned}
$$

We identify

$$\|\Delta_t\| = \max_{s'} \max_{c'} \max_{a'} |Q_t(s', a', c') - Q^*(s', a', c')|, \kappa = \gamma$$

and

$$c_t = \gamma \max_{s'} \max_{c'} \left| \sum_{a'} \pi^e_t(s', c', a') Q_t(s', a', c') - \max_{a'} Q_t(s', a', c') \right|.$$

The left is to verify c_t converges to zero w.p.1. Given the estimation policy is greedy in the limit, that is,

$$\lim_{t \to \infty} \sum_{a} \pi_t(s, c, a) Q_t(s, c, a) = \max_{a} Q_t(s, a, c), \forall s, c,$$

we have

$$
\begin{aligned}
&\lim_{t \to \infty} \gamma \max_{s'} \max_{c'} \left| \sum_{a'} \pi^e_t(s', c', a') Q_t(s', a', c') - \max_{a'} Q_t(s', a', c') \right| \\
=&\gamma \max_{s'} \max_{c'} \left| \lim_{t \to \infty} \sum_{a'} \pi^e_t(s', c', a') Q_t(s', a', c') - \max_{a'} Q_t(s', a', c') \right| \\
=&0
\end{aligned}
$$

□

C. Hyper parameters and Implementation Details