DynaMITE-RL: A Dynamic Model for Improved Temporal Meta-Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce *DynaMITE-RL*, a meta-reinforcement learning (meta-RL) approach to approximate inference in environments where the latent state evolves at varying rates. We model episode sessions—parts of the episode where the latent state is fixed—and propose three key modifications to existing meta-RL methods: (i) consistency of latent information within sessions, (ii) session masking, and (iii) prior latent conditioning. We demonstrate the importance of these modifications in various domains, ranging from discrete Gridworld environments to continuous-control and simulated robot assistive tasks, illustrating the efficacy of DynaMITE-RL over state-of-the-art baselines in both online and offline RL settings.

1 Introduction

Markov decision processes (MDPs) [4] provide a general framework in reinforcement learning (RL), and can be used to model sequential decision problems in a variety of domains, e.g., recommender systems (RSs), robot and autonomous vehicle control, and healthcare [22, 21, 7, 46, 31, 5]. MDPs assume a static environment with fixed transition probabilities and rewards [3]. In many real-world systems, however, the dynamics of the environment are intrinsically tied to latent factors subject to temporal variation. While non-stationary MDPs are special instances of partially observable MDPs (POMDPs) [24], in many applications these latent variables change infrequently, i.e. the latent variable remains fixed for some duration before changing. One class of problems exhibiting this latent transition structure is recommender systems, where a user's preferences are a latent variable which gradually evolves over time [23, 26]. For instance, a user may initially have a strong affinity for a particular genre (e.g., action movies), but their viewing habits could change over time, influenced by external factors such as trending movies, mood, etc. A robust system should adapt to these evolving tastes to provide suitable recommendations. Another example is in manufacturing settings, where industrial robots may experience unobserved gradual deterioration of their mechanical components affecting the overall functionality of the system. Accurately modelling such latent transitions caused by hardware degradation can help manufacturers optimize performance, cost, and equipment lifespan.

Our goal in this work is to leverage such a temporal structure to obviate the need to solve a fully general POMDP. To this end, we propose **Dyna**mic **M**odel for **I**mproved **T**emporal **M**e**t**a **R**einforcement **L**earning (DynaMITE-RL), a method designed to exploit the temporal structure of sessions, i.e., sub-trajectories within the history of observations in which the latent state is fixed. We formulate our problem as a *dynamic latent contextual MDP (DLCMDP)*, and identify three crucial elements needed to enable tractable and efficient policy learning in environments with the latent dynamics captured by a DLCMDP. First, we consider consistency of latent information, by exploiting time steps for which we have high confidence that the latent variable is constant. To do so, we introduce a consistency loss to regularize the posterior update model, providing better posterior estimates of the latent variable. Second, we enforce the posterior update model to learn the dynamics of the latent variable. This

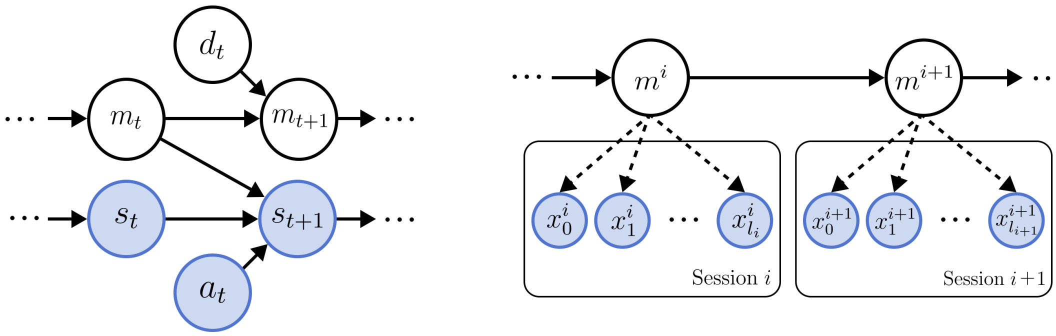

Figure 1: **(Left)** The graphical model for a DLCMDP. The transition dynamics of the environment follows $T(s_{t+1}, m_{t+1} \mid s_t, a_t, m_t)$. At every timestep t, an i.i.d. Bernoulli random variable, d_t, denotes the change in the latent context, m_t. Blue shaded variables are observed, whereas white shaded variables are latent. **(Right)** A realization of a DLCMDP episode. Each session i is governed by a latent variable m^i which is changing between sessions according to a fixed transition function, $T_m(m' \mid m)$. We denote l_i as the length of session i. The state-action pair (s_t^i, a_t^i) at timestep t in session i is summarized into a single observed variable, x_t^i. We emphasize that session terminations are not explicitly observed.

allows the trained policy to better infer, and adapt to, temporal shifts in latent context in unknown environments. Finally, we show that the variational objective in meta-RL algorithms, which attempts to reconstruct the entire trajectory, can hurt performance when the latent context is nonstationary. We modify this objective to reconstruct only the transitions that share the same latent context.

Closest to our work is VariBAD [47], a meta-RL [1] approach for learning a Bayes-optimal policy, enabling an agent to quickly adapt to a new environment with unknown dynamics and reward functions. VariBAD uses variational inference to learn a posterior update model that approximates the belief over the distribution of transition and reward functions. It augments the state space with this belief to encode the agent's uncertainty during decision-making. Nevertheless, VariBAD and the Bayes-Adaptive MDP framework [35] assume the latent context is static *across an episode* and do not address settings with latent state dynamics. In this work, we focus on the dynamic latent state formulation of the meta-RL problem.

Our core contributions are as follows: (1) We introduce DynaMITE-RL, a meta-RL approach to handle environments with evolving latent context variables. (2) We introduce three key elements for learning an improved posterior update model: session consistency, modeling dynamics of latent context, and session reconstruction masking. (3) We validate our approach on a diverse set of challenging simulation environments and demonstrate significantly improved results over multiple state-of-the-art baselines in both online and offline-RL settings.

2 Background

We begin by reviewing relevant background including meta-RL and Bayesian RL. We also briefly summarize the VariBAD [47] algorithm for learning Bayes-adaptive policies.

Meta-RL. The goal of meta-RL [1] is to quickly adapt an RL agent to an unseen test environment. Meta-RL assumes a distribution $p(\mathcal{T})$ over possible environments or *tasks*, and learns this distribution by repeatedly sampling batches of tasks during meta-training. Each task $\mathcal{T}_i \sim p(\mathcal{T})$ is described by an MDP $\mathcal{M}_i = (\mathcal{S}, \mathcal{A}, R_i, T_i, \gamma)$, where the state space $\mathcal{S}$, action space $\mathcal{A}$, and discount factor γ are shared across tasks, while R_i and T_i are task-specific reward and transition functions, respectively. The objective of meta-RL is to learn a policy that efficiently maximizes reward given a new task $\mathcal{T}_i \sim p(\mathcal{T})$ sampled from the task distribution at meta-test time. Meta-RL is a special case of a POMDP in which the unobserved variables are R and T, which are assumed to be stationary throughout an episode.

Bayesian Reinforcement Learning (BRL). BRL [18] utilizes Bayesian inference to model the uncertainty of agent and environment in sequential decision making problems. In BRL, R and T are unknown a priori and treated as random variables with associated prior distributions.

At time t, the *observed history* of states, actions and rewards is $\tau_{:t} = \{s_0, a_0, r_1, \dots, r_t, s_t\}$, and the belief b_t represents the posterior over task parameters R and T given the transition history, i.e. $b_t \triangleq p(R, T \mid \tau_{:t})$. Given the initial belief $b_0(R, T)$, the belief can be updated iteratively using Bayes' rule: $b_{t+1} = p(R, T \mid \tau_{:t+1}) \propto p(s_{t+1}, r_{t+1} \mid \tau_{:t}, R, T) \cdot b_t$. This Bayesian approach to RL can be formalized as a *Bayes-adaptive MDP (BAMDP)* [14]. A BAMDP is an MDP over the *augmented state space* $S^+ = \mathcal{S} \times \mathcal{B}$, where $\mathcal{B}$ denotes the belief space. Given the augmented state $s_t^+ = (s_t, b_t)$, the transition function is given by $T^+(s_{t+1}^+ \mid s_t^+, a_t) = \mathbb{E}_{b_t}[T(s_{t+1} \mid s_t, a_t) \cdot \delta(b_{t+1} = p(R, T \mid \tau_{:t+1})]$, and reward function is the expected reward given the belief, $R^+(s_t^+, a_t) = \mathbb{E}_{b_t}[R(s_t, a_t)]$. The BAMDP formulation naturally resolves the exploration-exploitation tradeoff. A Bayes-optimal RL agent takes information-gathering actions to reduce its uncertainty in the MDP parameters while simultaneously maximizing its returns. However, for most interesting problems, solving the BAMDP—and even computing posterior updates—is intractable given the continuous and typically high-dimensional nature of its state space.

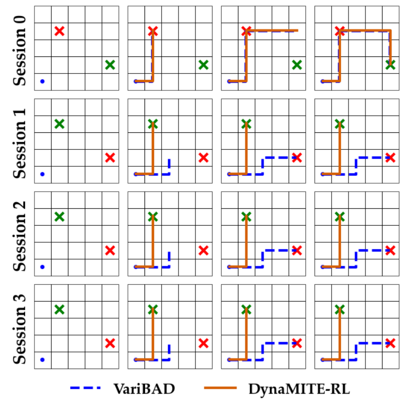

Figure 2: A DLCMDP rollout. VariBAD does not model the transition dynamics of the latent context and fails to adapt to the changing goal location. By contrast, DynaMITE-RL correctly infers the transition and consistently reaches the rewarding cell (green cross).

VariBAD. Zintgraf et al. [47] approximates the Bayes-optimal solution by modeling uncertainty over the MDP parameters. These parameters are represented by a latent vector $m \in \mathbb{R}^d$, the posterior over which is $p(m \mid \tau_{:H})$, where H is the BAMDP horizon. VariBAD uses a variational approximation $q_\phi(m \mid \tau_{:t})$ parameterized by ϕ and conditioned on the observed history up to time t. Zintgraf et al. [47] show that $q_\phi(m \mid \tau_{:t})$ approximates the belief b_t. In practice, $q_\phi(m \mid \tau_{:t})$ is represented by a Gaussian distribution $q_\phi(m \mid \tau_{:t}) = \mathcal{N}(\mu(\tau_{:t}), \Sigma(\tau_{:t}))$, where μ and Σ are sequence models (e.g., recurrent neural networks or transformers [42]) that encode trajectories to latent statistics. The variational lower bound at time t is $\mathbb{E}_{q_\phi(m|\tau_{:t})}[\log p_\theta(\tau_{:H} \mid m)] - D_{KL}(q_\phi(m \mid \tau_{:t}) \parallel p_\theta(m))$, where the first term reconstructs the trajectory likelihood $p_\theta(\tau_{:H} \mid m)$ and the second term regularizes the variational posterior to a prior distribution over the latent space, typically modeled with a standard Gaussian distribution. Importantly, the trajectory up to time t, i.e., $\tau_{:t}$, is used in the ELBO equation to infer the posterior belief at time t, which then decodes the entire trajectory $\tau_{:H}$, *including future transitions*. Given the belief state distribution q_ϕ of a BAMDP, the policy maps both the state and belief to actions, i.e., $\pi(a_t \mid s_t, q_\phi(m \mid \tau_{:t}))$. The BAMDP solution policy π^* is trained, e.g., via policy gradient methods, to maximize the expected cumulative return of meta-RL: $J(\pi) = \mathbb{E}_{R,T}\left[\mathbb{E}_\pi[\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t)]\right]$, where the first expectation is averaged over environments. The RL agent is trained jointly with the variational belief distribution q_ϕ.

3 Dynamic Latent Contextual MDPs

As a special case of a BAMDP, where the belief state is parameterized with a latent context vector (analogous to the problem formulation of VariBAD), the *dynamic latent contextual MDP (DLCMDP)* is denoted by $\langle \mathcal{S}, \mathcal{A}, \mathcal{M}, R, T, \nu_0, H \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{M}$ is the *latent* context space, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{M} \mapsto \Delta_{[0,1]}$ is a reward function, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{M} \mapsto \Delta_{\mathcal{S} \times \mathcal{M}}$ is a transition function, $\nu_0 \in \Delta_{\mathcal{S} \times \mathcal{M}}$ is an initial state distribution, $\gamma \in (0, 1)$ is a discount factor, and H is the (possibly infinite) horizon.

We assume an episodic setting in which each episode begins in a state-context pair $(s_0, m_0) \sim \nu_0$. At time t, the agent is at state s_t and context m_t, and has observed history $\tau_{:t} = \{s_0, a_0, r_1, \dots, r_t, s_t\}$. Given the history, the agent selects an action $a_t \in \mathcal{A}$, after which the state and latent context transitions according to $T(s_{t+1}, m_{t+1} \mid s_t, a_t, m_t)$, and the agent receives a reward sampled from $R(s_t, a_t, m_t)$. Throughout this process, the context m_t is latent (i.e., *not observed* by the agent).

DLCMDPs embody the causal independence depicted by the graphical model in Figure 1. Particularly, DLCMDPs impose a structure on changes of the latent variable m, allowing the latent context m to change less or more frequently. We denote by d_t the random variable at which a transition occurs in

DynaMITE-RL Training

```
Input: env, policy, critic, belief model
for iter = 1 to num_rl_updates do
    Collect DLCMDP episodes
    Train posterior belief model by maximizing
    ELBO (Eq. (2))
    Train policy and critic with any online RL
    algorithm
end for
```

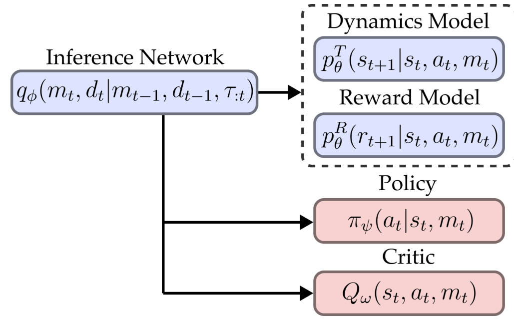

Figure 3: Pseudo-code (online RL training) and model architecture of DynaMITE-RL.

m_t. Let $\Omega = \{d_t\}_{t=0}^{H-1}$ denote a sequence of i.i.d. Bernoulli random variables, according to Figure 1, the transition function T is represented by the following factored distribution:

$$\begin{aligned} &T(s_{t+1} = s', m_{t+1} = m' \mid s_t = s, a_t = a, m_t = m) \\ &= T_s(s' \mid s, a, m)\mathbb{1}\{m' = m, d_t = 0\}T_d(d_t = 0) + \nu_0(s' \mid m')T_m(m' \mid m)\mathbb{1}\{d_t = 1\}T_d(d_t = 1), \end{aligned}$$

where $T_m : \mathcal{M} \mapsto \mathcal{M}$ is the latent dynamics function, T_s is the context-dependent state transition function, and T_d is the termination probability distribution. We refer to sub-trajectories between changes in the latent context as *sessions*, which may vary in length. At the start of a new session, a new state and a new latent context are sampled based on the distribution ν_0. Each session itself is governed by an MDP parameterized with a latent context $m \in \mathcal{M}$, which changes stochastically between sessions according to the latent transition function $T_m(m' \mid m)$. For notational simplicity we use index i to denote the i^{th} session in a trajectory, and m^i the respective latent context of that session. We emphasize that sessions switching times are latent random variables.

Notice that DLCMDPs are more general than latent MDPs [38, 29], in which the latent context is fixed throughout the entire episode; this corresponds to $d_t \equiv 0$. Moreover, DLCMDPs are closely related to POMDPs; letting $d_t \equiv 1$, a DLCMDP reduces to a general POMDP with state space $\mathcal{M}$, observation space $\mathcal{S}$, and observation function ν_0. As a consequence DLCMDPs are as general as POMDPs, rendering them very expressive. Moreover, the specific temporal structure of DLCMDPs allows us to devise efficient learning algorithms that exploit the transition dynamics of the latent context, improving learning efficiency. DLCMDPs are related to DCMDPs [40], LSMDPs [8], and DP-MDP [45]. However, DCMDPs assume contexts are observed, and focus on aggregated context dynamics, LSMDPs assume that the latent contexts across sessions are i.i.d (i.e., there is no latent dynamics) and DP-MDPs assume that sessions are fixed length.

We aim to learn a policy $\pi(a_t \mid s_t, m_t)$ which maximizes the expected return $J(\pi)$ over unseen test environments. As in BAMDPs, the optimal DLCMDP Q-function satisfies the Bellman equation; $\forall s^+ \in \mathcal{S}^+, a \in \mathcal{A} : Q(s^+, a) = R^+(s^+, a) + \gamma \sum_{s^{+\prime} \in \mathcal{S}^+} T^+(s^{+'} \mid s^+, a) \max_{a'} Q(s^{+'}, a)$. In the following section, we present DynaMITE-RL for learning a Bayes-optimal agent in a DLCMDP.

4 DynaMITE-RL

We detail DynaMITE-RL, first deriving a variational lower bound for learning a DLCMDP posterior model, then outlining three principles for training DLCMDPs, and finally integrating them into our training objective.

Variational Inference for Dynamic Latent Contexts. Given that we do not have direct access to the transition and reward functions of the DLCMDP, following Zintgraf et al. [47], we infer the posterior $p(m \mid \tau_{:t})$, and reason about the latent context vector m instead. Since exact posterior computation over m is computationally infeasible, given the need to marginalize over task space, we introduce the variational posterior $q_\phi(m \mid \tau_{:t})$, parameterized by $\phi \in \mathbb{R}^d$, to enable fast inference at every step. Our learning objective maximizes the log-likelihood $\mathbb{E}_\pi[\log p(\tau)]$ of observed trajectories. In general, the true posterior over the latent context is intractable, as is the empirical estimate of the log-likelihood. To circumvent this, we derive the *evidence lower bound (ELBO)* [27] to approximate the posterior over m under the variational inference framework.

Let $\mathcal{Z} = \{m^i\}_{i=0}^{K-1}$ be the sequence of latent context vectors for K sessions in an episode (note that K is inherently a random variable—the exact number of sessions in an episode is not known). As defined previously, Ω is the collection of the session terminations. We use a parametric generative distribution model for the state-reward trajectory, conditioned on the action sequence: $p_\theta(s_0, r_1, s_1, \ldots, r_H, s_H \mid a_0, \ldots, a_{H-1})$. In what follows, we drop the conditioning on $a_{:H-1}$ for the sake of brevity.

The variational lower bound can be expressed as:

$$\log p_\theta(\tau) \geq \underbrace{\mathbb{E}_{q_\phi(\mathcal{Z},\Omega|\tau_{:t})}\big[\log p_\theta(\tau \mid \mathcal{Z}, \Omega)\big]}_{\text{reconstruction}} - \underbrace{D_{KL}(q_\phi(\mathcal{Z}, \Omega \mid \tau_{:t})) \parallel p_\theta(\mathcal{Z}, \Omega))}_{\text{regularization}} = \mathcal{L}_{\text{ELBO},t}, \quad (1)$$

which can be estimated via Monte Carlo sampling over a learnable approximate posterior q_ϕ. In optimizing the reconstruction loss of session transitions and rewards, the learned latent variables should capture the unobserved MDP parameters. The full derivation of the ELBO for a DLCMDP is provided in Appendix A.1.

Figure 2 depicts a (qualitative) didactic GridWorld example with two possible rewarding goals that alternate between sessions. The VariBAD agent does not account for latent goal dynamics and gets stuck after reaching the goal in the first session. By contrast, DynaMITE-RL employs the latent context dynamics model to capture goal changes, and adapts to the context changes across sessions.

Consistency of Latent Information. In the DLCMDP formulation, each session is itself an MDP with a latent context fixed across the session. This within-context stationarity means new observations can only increase the information the agent has about this context. In other words, the agent's posterior over latent contexts gradually hone in on the true latent distribution. Although this true distribution remain unknown, this insight suggest the use of a *session-based consistency loss*, which penalizes an increase in KL-divergence between the current and final posterior belief within a session. Let $d_{H-1} = 1$ and $t_i \in \{0, \ldots, H\}$ be a random variable denoting the last timestep of session $i \in \{0, \ldots, K-1\}$, i.e., $t_i = \min\{t' \in \mathbb{Z}_{\geq 0} : \sum_{t=0}^{t'} d_t = i + 1\}$. At each time t in session i, we define the temporal, session-based consistency loss as

$$\mathcal{L}_{\text{consistency},t} = \max\{D_{KL}(q_\phi(m^i \mid \tau_{:t+1}) \parallel q_\phi(m^i \mid \tau_{:t_i})) - D_{KL}(q_\phi(m^i \mid \tau_{:t}) \parallel q_\phi(m^i \mid \tau_{:t_i})),\ 0\},$$

where $q_\phi(m^i \mid \tau_{:t_i})$ is the final posterior in session i. Using temporal consistency to regularize inference introduces an explicit inductive bias that allows for better posterior estimation.

Remark 4.1. We introduce session-based consistency for DLCMDPs, though it is also relevant in single-session settings with non-dynamic latent context. Indeed, as we discuss below, while VariBAD focuses on single sessions, it does not constrain the latent's posterior to be identical to final posterior belief. Consistency may be useful in settings where the underlying latent variable is stationary, but may hurt performance when this variable is indeed changing. Since our modeling approach allows latent context changes across sessions, incorporating consistency regularization does not generally hurt performance.

Latent Belief Conditioning. Unlike the usual BAMDP framework, DLCMDPs allow one to model temporal changes of latent contexts via dynamics $T_m(m' \mid m)$ across sessions. To incorporate this model into belief estimation, in addition to the history $(\tau_{:t}, d_{:t})$, we condition the posterior on the final latent belief $q_\phi(m', d' \mid m, d, \tau_{:t})$ from the previous session, and impose KL-divergence matching between this belief and the prior distribution $p_\theta(m' \mid m)$.

Reconstruction Masking. When the agent is at time t, Zintgraf et al. [47] encode past interactions to obtain the current posterior $q_\phi(m \mid \tau_{:t})$ since this is all the information available for inference about the current task (see Eq. (1)). They use this posterior to decode the entire trajectory—*including future transitions*—from different sessions to optimize the lower bound during training. The insight is that decoding both the past and future allows the posterior model to perform inference about unseen states. However, we observe that when the latent context is stochastic, reconstruction over the full sequence is detrimental to training efficiency. The model is attempting to reconstruct transitions outside of the current session that may be irrelevant or biased given the latent-state dynamics, rendering it a more difficult learning problem. Instead we reconstruct only the transitions within the session defined by the predicted termination indicators, i.e., at any arbitrary time t within session i, the session-based reconstruction loss is given by

$$\mathcal{L}_{\text{session-ELBO},t} = \mathbb{E}_{q_\phi(\mathcal{Z},\Omega|\tau_{:t})}\big[\log p_\theta(\tau_{t_{i-1}+1:t_i} \mid \mathcal{Z}, \Omega)\big] - D_{KL}(q_\phi(\mathcal{Z}, \Omega \mid \tau_{:t})) \parallel p_\theta(\mathcal{Z}, \Omega)).$$

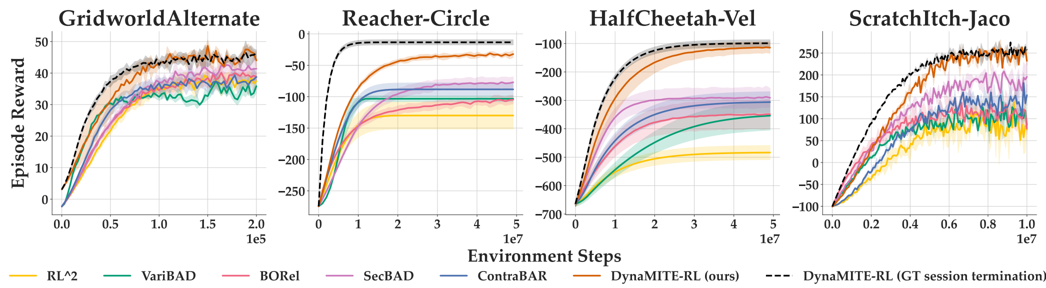

Figure 4: Learning curves for DynaMITE-RL and state-of-the-art baseline methods. Shaded areas represent standard deviation over 5 different random seeds for each method and 3 for ScratchItch. In each of the evaluation environments, we observe that DynaMITE-RL exhibits better sample efficiency and converges to a policy with better environment returns than the baseline methods.

DynaMITE-RL. By incorporating the three modifications above, we obtain at the following training objective for our variational meta-RL approach:

$$\mathcal{L}_{\text{DynaMITE-RL}}(\theta, \phi) = \sum_{t=0}^{H-1} \left[\mathcal{L}_{\text{session-ELBO},t}(\theta, \phi) + \beta \cdot \mathcal{L}_{\text{consistency},t}(\phi) \right], \tag{2}$$

where $\beta > 0$ is a hyper-parameter that regularizes the consistency loss. We present a simplified pseudocode for online training of DynaMITE-RL in Algorithm 3a and a detailed algorithm in Appendix A.2.

Implementation Details. We use proximal policy optimization (PPO) [37] for online RL training. We introduce a posterior inference network that outputs a Gaussian over the latent context for the i-th session and the session termination indicators, $q_\phi(m^i, d_{:t} \mid \tau_{:t}, m^{i-1})$, conditioned on the history and posterior belief from the previous session. We parameterize the inference network as a sequence model, with e.g., an RNN [9] or a Transformer [42], with different multi-layer perceptron (MLP) output heads for predicting the logits for session termination and the posterior belief. In practice, the posterior MLP outputs the parameters of a Gaussian belief distribution $q_{\phi_m}(m^i \mid \tau_{:t}, m^{i-1}) = \mathcal{N}(\mu(\tau_{:t}), \Sigma(\tau_{:t}))$. The session termination network applies a sigmoid activation function $\sigma(x) = \frac{1}{1+e^{-x}}$ to the MLP output. Following PPO [37], the actor loss $\mathcal{J}_\pi$ and critic loss $\mathcal{J}_\omega$ are respectively given by $\mathcal{J}_\pi = \mathbb{E}_{\tau \sim \pi_\psi}[\log \pi_\psi(a \mid s, m)\hat{A}(s, a, m)]$ and $\mathcal{J}_\omega = \mathbb{E}_{\tau \sim \pi_\psi}[(Q_\omega(s, a, m) - (r + V_\omega(s', m))^2]$, where V is the target network, and $\hat{A}$ is the advantage function. We also add an entropy bonus to ensure sufficient exploration in more complex domains. A decoder network, also parameterized using MLPs, reconstructs transitions and rewards given the session's latent context m^i, current state s_t, and action a_t, i.e., $p_\theta^T(s_{t+1} \mid s_t, a_t, m_t)$ and $p_\theta^R(r_{t+1} \mid s_t, a_t, m_t)$. Figure 3b depicts the implemented model architecture. The final objective of DLCMDP is to jointly learn the policy π_ψ, the variational posterior model q_ϕ, and the factored likelihood model p_θ that minimizes the following loss:

$$\mathcal{L}(\theta, \phi, \psi) = \mathbb{E}\left[\mathcal{J}_\pi(\psi) + \lambda \cdot \mathcal{L}_{\text{DynaMITE-RL}}(\phi, \theta) \right], \tag{3}$$

where $\mathcal{J}$ is the expected return, and $\lambda > 0$ is a hyper-parameter trades off this return with DynaMITE-RL's variational inference objective. We also evaluate DynaMITE-RL in an offline RL setting, in which we collect an offline dataset of trajectories following an oracle goal-conditioned policy and subsequently approximate the optimal value function and RL agent using offline RL methods, e.g., IQL [28]. The value function and the policy are parameterized with the same architecture as in the online setting and will be detailed in Appendix A.5.

5 Experiments

We present experiments that demonstrate, while VariBAD and other meta-RL methods struggle to learn good policies given nonstationary latent contexts, DynaMITE-RL exploits the causal structure of a DLCMDP to more efficiently learn performant policies. We compare our approach to several state-of-the-art meta-RL baselines, showing its significantly better evaluation returns.

Environments. We test DynaMITE-RL on a suite of standard meta-RL benchmark tasks including a didactic gridworld navigation, continuous control, and human-in-the-loop robot assistance as shown in Figure 8. Gridworld navigation and MuJoCo [41] locomotion tasks are considered by Zintgraf et al. [47], Dorfman et al. [12], and Choshen and Tamar [10]. We modify these environments to incorporate temporal shifts in the reward and/or environment dynamics. To achieve good performance under these conditions, a learned policy must adapt to the latent state dynamics. More details about the environments and hyperparameters can be found in Appendix A.4 and A.5.

Gridworld. We modify the Gridworld environment used by Zintgraf et al. [47]. In a 5×5 gridworld, two possible goals are sampled uniformly at random in each episode. One of the two goals has a $+1$ reward while the other has 0 reward. The rewarding goal location changes after each session according to a predefined transition function. Goal locations are provided to the agent in the state—the only latent information is which goal has positive reward.

Continuous Control. We experiment with two tasks from OpenAI Gym [6]: Reacher and HalfCheetah. Reacher is a two-jointed robot arm tasked with reaching a 2D goal location that moves along a circular path according to some unknown transition function. HalfCheetah is a locomotion task which we modify to incorporate changing latent contexts w.r.t. the target direction (HalfCheetah-Dir), target velocity (HalfCheetah-Vel), and target velocity with opposing wind forces (HalfCheetah-Wind+Vel).

Assistive Itch Scratching. Assistive Itch Scratch is part of the Assistive-Gym benchmark [15] consisting of a human and a wheelchair-mounted 7-degree-of-freedom (DOF) Jaco robot arm. The human has limited-mobility and requires robot assistance to scratch an itch. We simulate stochastic latent context by moving the itch location—unobserved by the agent—along the human's right arm.

Meta-RL Baselines. We compare DynaMITE-RL to several state-of-the-art (approximately) Bayes-optimal meta-RL methods including RL^2 [13], VariBAD [47], BORel [12], SecBAD [8], and ContraBAR [10]. RL^2 [13] is an RNN-based policy gradient method which encodes environment transitions in the hidden state and maintains them across episodes. VariBAD reduces to RL^2 without the decoder and the variational reconstruction objective for environment transitions. BORel primarily investigates offline meta-RL (OMRL) and proposes a few modifications such as reward relabelling to address the identifiability issue in OMRL. Chen et al. [8] proposes the latent situational MDP (LS-MDP), in which there is non-stationary latent contexts that are sampled i.i.d., and SecBAD, an algorithm for learning in an LS-MDP. However, they do not consider latent dynamics which a crucial aspect in many applications. ContraBAR employs a contrastive learning objective to discriminate future observations from negative samples to learn an *approximate* sufficient statistic of the history. As Zintgraf et al. [47] already demonstrate better performance by VariBAD than posterior sampling methods (e.g., PEARL [34]) we exclude such methods from our comparison.

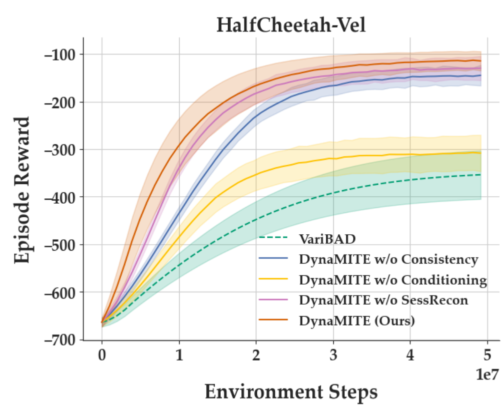

Figure 5: Ablating components of `DynaMITE-RL`. We observe that modelling latent dynamics is crucial in achieving good performance in a DLCMDP. Additionally, consistency regularization and session reconstruction improve the sample efficiency and convergence to a better performing policy.

DynaMITE-RL outperforms prior meta-RL methods in a DLCMDP in both online and offline RL settings. In Figure 4, we show the learning curves for DynaMITE-RL and baseline methods. We first observe that `DynaMITE-RL` significantly outperforms the baselines across all domains in sample efficiency and average environment returns. `RL`^{2}, `VariBAD`, `BORel`, `SecBAD`, and `ContraBAR` all perform poorly in the DLCMDP, converging to a suboptimal policy. By contrast, `DynaMITE-RL` accurately models the latent dynamics and consistently achieves high rewards despite the nonstationary latent context. We also evaluate an oracle with access to ground-truth session terminations and find that `DynaMITE-RL` with learned session terminations effectively recovers session boundaries and

Table 1: Average single episode returns for DynaMITE-RL and other state-of-the-art meta-RL algorithms across different environments. Results for all environments are averaged across 5 seeds beside ScratchItch which has 3 seeds. DynaMITE-RL, in bold, achieves the highest return on all of the evaluation environments and is the only method able to recover an optimal policy.

	Gridworld	Reacher	HC-Dir	HC-Vel	Wind+Vel	ScratchItch
RL^2	$33.4_{\pm 1.6}$	$-150.6_{\pm 1.2}$	$-420.0_{\pm 8.4}$	$-513.2_{\pm 8.7}$	$-493.5_{\pm 1.8}$	$50.4_{\pm 16.8}$
VariBAD	$31.8_{\pm 1.9}$	$-102.4_{\pm 4.2}$	$-242.5_{\pm 4.8}$	$-363.5_{\pm 3.2}$	$-188.5_{\pm 4.4}$	$81.8_{\pm 6.9}$
BORel	$32.4_{\pm 2.4}$	$-103.5_{\pm 4.6}$	$-240.6_{\pm 4.3}$	$-343.4_{\pm 3.6}$	$-167.8_{\pm 5.4}$	$82.5_{\pm 6.0}$
SecBAD	$38.5_{\pm 3.1}$	$-96.2_{\pm 4.8}$	$-202.4_{\pm 10.4}$	$-323.5_{\pm 3.4}$	$-155.3_{\pm 5.4}$	$101.4_{\pm 9.2}$
ContraBAR	$34.5_{\pm 0.9}$	$-101.6_{\pm 3.2}$	$-256.5_{\pm 3.6}$	$-312.3_{\pm 4.8}$	$-243.4_{\pm 2.6}$	$114.6_{\pm 24.4}$
DynaMITE-RL	$\mathbf{42.9}_{\pm 0.5}$	$\mathbf{-8.4}_{\pm 5.1}$	$\mathbf{-68.5}_{\pm 2.3}$	$\mathbf{-146.0}_{\pm 8.1}$	$\mathbf{-42.8}_{\pm 6.9}$	$\mathbf{231.2}_{\pm 23.3}$

Table 2: Average single episode returns with Offline RL. Results are averaged across 5 random seeds. Algorithm with the highest average return are shown in bold. We present results for an oracle agent trained with goal information for reference.

	Gridworld	Reacher	HC-Dir	HC-Vel	HC-Dir+Vel	ScratchItch
BORel	$31.4_{\pm 3.5}$	$-102.0_{\pm 5.8}$	$-245.0_{\pm 12.4}$	$-354.0_{\pm 8.3}$	$-170.0_{\pm 5.4}$	$72.5_{\pm 4.6}$
w/o Consistency	$38.2_{\pm 1.2}$	$-33.2_{\pm 2.7}$	$-206.0_{\pm 5.6}$	$-212.0_{\pm 6.4}$	$-120.0_{\pm 12.4}$	$105.8_{\pm 8.5}$
w/o Sess. Dynamics	$33.4_{\pm 1.3}$	$-95.0_{\pm 5.2}$	$-244.0_{\pm 6.0}$	$-342.0_{\pm 8.6}$	$-166.0_{\pm 9.5}$	$74.1_{\pm 2.3}$
DynaMITE-RL	$41.8_{\pm 0.6}$	$-15.5_{\pm 3.2}$	$-154.0_{\pm 8.6}$	$-156.0_{\pm 4.8}$	$-48.0_{\pm 8.6}$	$225.5_{\pm 10.6}$
w/ Transformer	$\mathbf{43.8}_{\pm 0.6}$	$\mathbf{-8.4}_{\pm 2.8}$	$\mathbf{-132.0}_{\pm 7.4}$	$\mathbf{-144.0}_{\pm 6.5}$	$\mathbf{-33.0}_{\pm 5.8}$	$\mathbf{242.5}_{\pm 7.4}$
Oracle (w/ goal)	44.6	-4.8	-112.0	-132.2	-24.4	245.3

matches oracle performance with sufficient training. Our empirical results validate that DynaMITE-RL learns a policy robust to changing latent contexts at inference time, while the baseline methods fail to adapt and get stuck in suboptimal behavior. We also demonstrate that DynaMITE-RL outperforms BORel in an offline RL setting in Table 2 in all environments. This highlights the importance of DynaMITE-RL training objectives in learning a more accurate posterior belief model even without online environment interactions. We also experimented with a Transformer encoder to parameterize our belief model and find that a more powerful model further improves the evaluation performance.

Each component of DynaMITE-RL contributes to efficient learning in a DLCMDP: We ablate the three key components of DynaMITE-RL to understand their impact on the resulting policy. We compare full DynaMITE-RL to: (i) DynaMITE-RL w/o Consistency, which does not include consistency regularization; (ii) DynaMITE-RL w/o Conditioning, which does not include latent conditioning; and (iii) DynaMITE-RL w/o SessRecon, which does not include session reconstruction. In Figure 5, we report the performance for each of these ablations and vanilla VariBAD for comparisons. First, without prior latent belief conditioning, the model converges to a suboptimal policy slightly better than VariBAD, confirming the importance of modeling the latent transition dynamics of a DLCMDP. Second, we find that session consistency regularization reinforces the inductive bias of changing dynamics and improves the sample efficiency of learning an accurate posterior model in DLCMDPs. Finally, session reconstruction masking also improves the sample efficiency by neglecting terms that are irrelevant and potentially biased. Similar ablation studies in the offline RL setting can be found in Table 2, reinforcing the importance of our proposed training objectives.

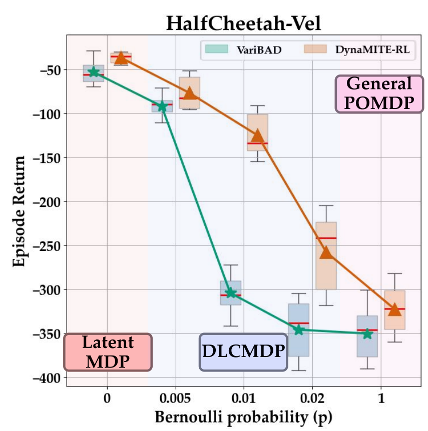

Figure 6: Ablation studies on various frequencies of latent context switches within an episode in the HalfCheetah-Vel environment. The boxplot shows the distribution over evaluation returns for 25 rollouts of trained policies with VariBAD and DynaMITE-RL . When $p = 0$, we have a latent MDP and when $p = 1$ this is equivalent to a general POMDP.

DynaMITE-RL is robust to varying levels of latent stochasticity. We study the effect of varying the number of latent context switches over an episode of fixed time horizon. For the HalfCheetah-Vel environment, we fix the episode horizon $H = 400$ to create multiple problems. We introduce a Bernoulli random variable, e.g $d_t \sim Bernoulli(p)$ where p is a hyperparameter we set to determine the probability that the latent context changes at timestep t. If $p = 0$, the latent context remains unchanged throughout the entire episode, corresponding to a latent MDP. If $p = 1$, the latent context changes at every timestep, which is equivalent to a general POMDP. As shown in Figure 6, DynaMITE-RL performs better, on average, than VariBAD, with lower variance in a latent MDP. We hypothesize that, in the case of latent MDP, consistency regularization helps learn a more accurate posterior model by enforcing the inductive bias that the latent is static. Otherwise, there is no inherent advantage in modeling the latent dynamics if it is stationary. As we gradually increase the number of context switches, the problem becomes more difficult and closer to a general POMDP. VariBAD performance decreases drastically because it is unable to model the changing latent dynamics while DynaMITE-RL is less affected, highlighting the robustness of our approach. When we set the number of contexts equal to the episode horizon length, we recreate a fully general POMDP and again the performance between VariBAD and DynaMITE-RL converges.

6 Related Work

POMDPs provide a general framework modeling non-stationality and partial observability in sequential decision problems. Many model variants have been introduced, defining a rich spectrum between episodic MDPs and POMDPs. The Bayes-adaptive MDP (BAMDP) [14] and hidden parameter MDP (HiP-MDP) [25] are both special cases of POMDPs in which environment parameters are unknown and the goal is to infer these parameters online during an episode. However, neither framework addresses the dynamics of the latent parameters across sessions, but rather assumes it is constant throughout an episode. LSMDP [8] and DP-MDP [44] do investigate nonstationary latent contexts but LSMDP samples them i.i.d., not considering the dynamics, while DP-MDP assumes fixed session lengths. By contrast, DLCMDPs models the dynamics of the latent state and simultaneously infers *when* the transition occurs, allowing better posterior updates at inference time.

DynaMITE-RL shares conceptual similarities with other meta-RL algorithms. Firstly, optimization-based techniques [16, 11, 36] learn neural network policies that can quickly adapt to new tasks at test time using policy gradient updates. However, these methods do not optimize for Bayes-optimal behavior and generally exhibit suboptimal test-time adaptation. Context-based meta-RL techniques aim to learn policies that directly infer task parameters at test time, conditioning the policy on the posterior belief. Such methods include recurrent memory-based architectures [13, 43, 30, 2] and variational approaches [20, 47, 12]. VariBAD, closest to our work, uses variational inference to approximate Bayes-optimal policies. However, we have demonstrated above the limitations of VariBAD in DLCMDPs, and have developed several crucial modifications to drive effective learning a highly performant policies in our setting.

7 Conclusion

We developed DynaMITE-RL, a meta-RL method to approximate Bayes-optimal behavior using a latent variable model. We presented the dynamic latent contextual Markov Decision Process (DLCMDP), a model in which latent context information changes according to an unknown transition function, that captures many natural settings. We derived a graphical model for this problem setting and formalized it as an instance of a POMDP. DynaMITE-RL is designed to exploit the causal structure of this model, and in a didactic GridWorld environment and several challenging continuous control tasks, we demonstrated that it outperforms existing meta-RL methods w.r.t. both learning efficiency and test-time adaptation in both online and offline-RL settings.

There are a number of exciting directions for future research building on the DLCMDP model. While we only consider Markovian latent dynamics in this work (i.e. future latent states are independent of prior latent states given the current latent state), we plan to investigate richer non-Markovian latent dynamics. We hope to extend DynaMITE-RL to other real-world applications including recommender systems (RS), autonomous driving, multi-agent collaborative systems, etc. DLCMDPs are a good model for RS as recommender agents often interact with users over long periods of time during which the user's latent context changes irregularly, directly influencing their preferences.

References

[1] J. Beck, R. Vuorio, E. Z. Liu, Z. Xiong, L. M. Zintgraf, C. Finn, and S. Whiteson. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*, 2023.

[2] J. Beck, R. Vuorio, Z. Xiong, and S. Whiteson. Recurrent hypernetworks are surprisingly strong in meta-rl. In *Advances in Neural Information Processing Systems*, 2023. URL `http://papers.nips.cc/paper_files/paper/2023/hash/c3fa3a7d50b34732c6d08f6f66380d75-Abstract-Conference.html`.

[3] R. Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, 6(5): 679—-84, 1957.

[4] D. Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific, 2012.

[5] E. Biyik, J. Margoliash, S. R. Alimo, and D. Sadigh. Efficient and safe exploration in deterministic markov decision processes with unknown transition models. In *American Control Conference*, pages 1792–1799. IEEE, 2019.

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

[7] Z. Cao, E. Bıyık, W. Z. Wang, A. Raventos, A. Gaidon, G. Rosman, and D. Sadigh. Reinforcement learning based control of imitative policies for near-accident driving. *Robotics: Science and Systems*, 2020.

[8] X. Chen, X. Zhu, Y. Zheng, P. Zhang, L. Zhao, W. Cheng, P. Cheng, Y. Xiong, T. Qin, J. Chen, et al. An adaptive deep rl method for non-stationary environments with piecewise stable context. *Neural Information Processing Systems*, 35:35449–35461, 2022.

[9] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014.

[10] E. Choshen and A. Tamar. Contrabar: Contrastive bayes-adaptive deep rl. In *International Conference on Machine Learning*, volume 202, pages 6005–6027, 2023.

[11] I. Clavera, J. Rothfuss, J. Schulman, Y. Fujita, T. Asfour, and P. Abbeel. Model-based reinforcement learning via meta-policy optimization. In *Conference on Robot Learning*, pages 617–629. PMLR, 2018.

[12] R. Dorfman, I. Shenfeld, and A. Tamar. Offline meta reinforcement learning–identifiability challenges and effective data collection strategies. *Neural Information Processing Systems*, 34: 4607–4618, 2021.

[13] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel. Rl^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

[14] M. O. Duff. *Optimal learning: computational procedures for bayes-adaptive markov decision processes*. PhD thesis, University of Massachusetts Amherst, 2002.

[15] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp. Assistive gym: A physics simulation framework for assistive robotics. In *IEEE International Conference on Robotics and Automation*. IEEE, 2020.

[16] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[17] C. D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem. Brax - a differentiable physics engine for large scale rigid body simulation. *arXiv preprint arXiv:2106.13281*, 2021. URL `http://github.com/google/brax`.

[18] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar. Bayesian reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 8(5-6):359–483, 2015.

[19] S. Huang, R. F. J. Dossa, A. Raffin, A. Kanervisto, and W. Wang. The 37 implementation details of proximal policy optimization. In *ICLR Blog Track*, 2022. URL `https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/`.

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

[21] E. Ie, C. Hsu, M. Mladenov, V. Jain, S. Narvekar, J. Wang, R. Wu, and C. Boutilier. RecSim: A configurable simulation platform for recommender systems. *arXiv preprint arXiv:1909.04847*, 2019.

[22] D. Jannach, A. Manzoor, W. Cai, and L. Chen. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)*, 54(5):1–36, 2021.

[23] G. Jawaheer, P. Weller, and P. Kostkova. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems*, 4(2):1–26, 2014.

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

[25] T. W. Killian, S. Daulton, G. Konidaris, and F. Doshi-Velez. Robust and efficient transfer learning with hidden parameter markov decision processes. *Neural Information Processing Systems*, 2017.

[26] C. Kim, J. Park, J. Shin, H. Lee, P. Abbeel, and K. Lee. Preference transformer: Modeling human preferences using transformers for rl. *International Conference of Learning Representations*, 2023.

[27] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[28] I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2021.

[29] J. Kwon, Y. Efroni, C. Caramanis, and S. Mannor. Rl for latent mdps: Regret guarantees and a lower bound. *Neural Information Processing Systems*, 34:24523–24534, 2021.

[30] G. Lee, B. Hou, A. Mandalika, J. Lee, S. Choudhury, and S. S. Srinivasa. Bayesian policy optimization for model uncertainty. *International Conference on Learning Representations*, 2018.

[31] S. Liu, K. C. See, K. Y. Ngiam, L. A. Celi, X. Sun, and M. Feng. Reinforcement learning for clinical decision support in critical care: comprehensive review. *Journal of Medical Internet Research*, 22(7):e18477, 2020.

[32] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[33] X. B. Peng, A. Kumar, G. Zhang, and S. Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

[34] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning*, pages 5331–5340. PMLR, 2019.

[35] S. Ross, B. Chaib-draa, and J. Pineau. Bayes-adaptive pomdps. *Neural Information Processing Systems*, 2007.

[36] J. Rothfuss, D. Lee, I. Clavera, T. Asfour, and P. Abbeel. Promp: Proximal meta-policy search. *International Conference on Learning Representations*, 2018.

[37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[38] L. N. Steimle, D. L. Kaufman, and B. T. Denton. Multi-model markov decision processes. *IISE Transactions*, 53(10):1124–1139, 2021.

[39] G. Tennenholtz, A. Hallak, G. Dalal, S. Mannor, G. Chechik, and U. Shalit. On covariate shift of latent confounders in imitation and reinforcement learning. *International Conference of Learning Representations*, 2022.

[40] G. Tennenholtz, N. Merlis, L. Shani, M. Mladenov, and C. Boutilier. Reinforcement learning with history dependent dynamic contexts. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 34011–34053. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/tennenholtz23a.html`.

[41] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Neural Information Processing Systems*, 30, 2017.

[43] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

[44] A. Xie and C. Finn. Lifelong robotic reinforcement learning by retaining experiences. In *Conference on Lifelong Learning Agents, CoLLAs 2022*, volume 199 of *Proceedings of Machine Learning Research*, pages 838–855, 2022.

[45] A. Xie, J. Harrison, and C. Finn. Deep reinforcement learning amidst continual structured non-stationarity. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11393–11403, 2021.

[46] C. Yu, J. Liu, S. Nemati, and G. Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

[47] L. Zintgraf, K. Shiarlis, M. Igl, S. Schulze, Y. Gal, K. Hofmann, and S. Whiteson. VariBAD: A very good method for bayes-adaptive deep rl via meta-learning. *International Conference of Learning Representations*, 2020.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Section 5 demonstrates, while VariBAD and other meta-RL methods struggle to learn good policies given nonstationary latent contexts, DynaMITE-RL exploits the causal structure of a DLCMDP to more efficiently learn performant policies in both online and offline-RL settings.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We only consider Markovian latent dynamics here (i.e. future latent states are independent of prior latent states given the current latent state). It would be interesting to explore complex non-Markovian latent dynamics.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not derive new theoretical results in this work.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We present all the information needed in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
    (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
    (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
    (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
    (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Not at this point but we will release the code along with the camera ready version of the paper. We will integrate several other meta-RL environments in addition to the ones discussed in the paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: As said, we present all the information needed in the Appendix. We disclose hyperparameters in Appendix A.5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Tables 1 and 2 have error bars. Figures 5 and 6 also have error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section A.5.2 provides information on the computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that this paper conforms the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper is about foundational research and not tied to particular applications currently. In the future, DynaMITE-RL can be used in assistive robots to improve healthcare delivery and patient satisfaction as we demonstrate in the experiments with Assistive Itch Scratch.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the creators of code used in the paper are credited and VariBAD, RL^2, BORel, SecBAD, and ContraBAR are under MIT License.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.