Inferring Behavior-Specific Context Improves Zero-Shot Generalization in Reinforcement Learning

Tidiane Camaret Ndir **André Biedenkapp** **Noor Awad**
{ndirt, biedenka, awad}@cs.uni-freiburg.de University of Freiburg

Abstract

In zero-shot generalization (ZSG) in Reinforcement Learning (RL) agents must adapt to entirely novel environments without additional training. Understanding and utilizing contextual cues, such as the gravity level of the environment, is critical for robust generalization. We propose to integrate the learning of context representations directly with policy learning. Our algorithm demonstrates improved generalization on various simulated domains, outperforming prior context-learning techniques in zero-shot settings. By jointly learning policy and context, our method develops behavior-specific context representations, enabling adaptation to unseen environments. This approach marks significant progress toward reinforcement learning systems that can generalize across diverse real-world tasks.

1 Introduction

Reinforcement learning [RL; Sutton and Barto, 2018] is a key area in creating intelligent systems that can learn and adapt to various tasks. A significant challenge in RL is crafting algorithms that perform well not only in known environments but also in new, unseen ones. This highlights the need for generalizable RL agents. Generalization in RL includes a range of definitions and strategies, each suited to the specific requirements of different applications. The concept of generalization in RL covers everything from basic domain adjustments to advanced multitask learning settings [Kirk et al., 2023].

In this context, zero-shot generalization (ZSG) is a notable aspect of generalization. ZSG is unique because it requires that the learned policies are able to be applied to new environments without additional modifications during testing [Kirk et al., 2023, Benjamins et al., 2023]. This is especially important for applications where it is not possible to fine-tune the model in the target environment. ZSG is attractive because it suggests that RL models can effectively handle the diversity and unpredictability of real-world tasks. However, RL algorithms often struggle with even minor changes in their environment, which can result in behaviors that do not transfer well [Henderson et al., 2018, Andrychowicz et al., 2021]. We attribute this issue primarily to the complexity of the RL training process, which typically does not focus on generalization [Parker-Holder et al., 2022].

The path to achieving reliable ZSG for RL presents many challenges. In particular, how can zero-shot generalization be achieved if we do not have access to privileged information, i.e. a context [Hallak et al., 2015, Modi et al., 2018], that can help us identify the underlying transition dynamics. For example, an RL agent assigned to control a robot may adjust its control behavior based on the weight of the load the robot needs to carry. If the robot is equipped with sensors that detect this weight, the RL policy could immediately use the information to adapt the control accordingly. However, if this information is not easily accessible, learning a policy that can adapt to such changes becomes even more difficult. For this reason, recent works have proposed inferring a context from previous observations [see, e.g., Zhou et al., 2019, Evans et al., 2022]. Although these works have shown great promise for zero-shot generalization, they often fall short of realizing their full potential. We argue that this is largely due to the decoupling of learning the context from learning the policy. Instead,

17th European Workshop on Reinforcement Learning (EWRL 2024).

here, we propose learning behavior-specific context to aid policy learning by jointly learning context representations and well-performing policies.

In summary, this paper delves into the intricate domain of generalization in model-free RL, with a focus on zero-shot generalization from an inferred context. Specifically, the contributions of our work are as follows:

- We propose a novel RL algorithm, capable of zero-shot generalization by learning to recognize the environment dynamics while jointly learning desirable behaviors within the environment.
- Our empirical evaluation of the proposed algorithm contrasts the learning behavior of RL agents when providing explicit access to a ground truth context versus an inferred context.
- We provide insights into the learned context embedding and show that they recover the relationship between transition dynamics.

2 Related Work

Various approaches have been proposed to learn generalizable RL agents. In this line of research, two particular areas have gained more momentum with a focus on either few-shot or zero-shot generalization abilities.

Meta-RL Meta-reinforcement learning [Meta-RL; Beck et al., 2023] involves the process of learning how to learn in reinforcement learning. Thus, the goal is to learn RL pipelines that can learn efficiently, so that they can be easily transferred to new settings and learn to solve a new task with few interactions. For example, Duan et al. [2016] proposed to encode the learning dynamics of a proximal policy optimization [PPO; Schulman et al., 2017] in a recurrent neural network (RNN). Thus, by giving examples of how the PPO agent adapts its behavior over time during training, the RNN can learn how to adapt a policy, without needing to perform gradient updates at test time. In an ideal scenario, such a learned RL agent only needs a few exploration episodes to find the optimal behavior, i.e., few-shot adaptation. To identify environment dynamics, most of such model free meta-RL agents [see, e.g., Finn et al., 2017, Wang et al., 2017, Rakelly et al., 2019, Nagabandi et al., 2019, Melo, 2022] and model-based ones [see, e.g., Lee et al., 2020, Guo et al., 2022, Sodhani et al., 2022, Wen et al., 2023] keep a short history of transitions to estimate environment transition dynamics. Although much progress has been reported in meta-RL, at test time, many thousands of environment interactions are required for the learned RL agents to solve the test tasks reliably.

Contextual RL In contextual RL (cRL), it is assumed that the underlying transition dynamics can be characterized by a context [Hallak et al., 2015, Modi et al., 2018]. This could, for example, be a physical property such as wind [Koppejan and Whiteson, 2009], the length of the pole that needs to be balanced [Seo et al., 2020, Kaddour et al., 2020, Benjamins et al., 2023], the characteristic of the terrain [Escontrela et al., 2020], or more abstract concepts that characterize the dynamics of the underlying environment [Biedenkapp et al., 2020, Adriaensen et al., 2022]. Kirk et al. [2023] identify the cRL setting as particularly relevant for the study of zero-shot generalization capabilities of RL agents, as the cRL framework allows us to define the ranges of inter- and extrapolation distributions and enable a systematic and principled study of how RL agents can adapt to changes in their environments. Using the evaluation protocol proposed by Kirk et al. [2023], Benjamins et al. [2023] studied the generalizability of multiple model-free RL agents on a benchmark that uses various physical properties as context information. Their study assumed a naive use of context information by concatenating it directly with the observation. Instead, Beukman et al. [2023] proposed to use a hypernetwork to learn adaptable RL agents. However, their approach requires that agents can directly observe the context. In contrast, our work studies the effectiveness of *inferring* context, as is usually done in meta-RL, for zero-shot generalization without assuming access to the context.

3 Background - Contextual Markov Decision Processes

We build on the framework of contextual Markov decision processes [cMDPs; Hallak et al., 2015, Modi et al., 2018], which was proposed as an ideal abstraction for the study of zero-shot generalization in RL [Kirk et al., 2023]. In an MDP $M = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho)$, the characteristics of the environment are

represented by the state space $\mathcal{S}$, the action space $\mathcal{A}$, the transition dynamics $\mathcal{T}$, the reward function $\mathcal{R}$, and the initial state distribution ρ.

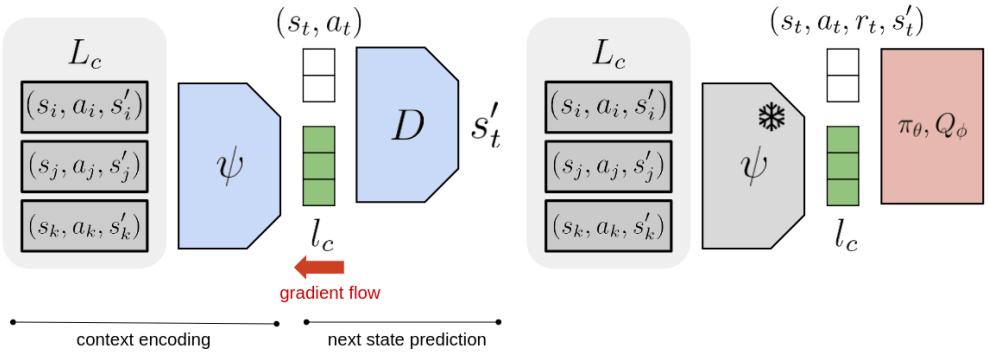
cMDPs introduce context to parameterize an environment's behavioral rules, allowing for variations in task instances. In a cMDP $\mathcal{M}_{\text{cMDP}}$, the action space $\mathcal{A}$ and the state space $\mathcal{S}$ remain untouched, while the transition dynamics $\mathcal{T}_{\text{c}}$, the reward $\mathcal{R}_{\text{c}}$ and the initial state distribution ρ_{c} are context dependent and vary with context $\text{c} \in C$. The context-aware initial state distribution ρ_{c} and dynamic changes expose the agent to different parts of the state space across contexts. Therefore, a cMDP $\mathcal{M}_{\text{cMDP}}$ encompasses a set of MDPs that vary by context, denoted as $\{\mathcal{M}_c\}_{\text{c}\in C}$, which provides a framework for studying generalization in RL in diverse environments.

In the context of zero-shot problems, we can then utilize cMDPs to study generalization by defining two sets of context sets C_{train} and C_{eval}, for training and evaluation, respectively [Kirk et al., 2023]. During training, the context values are sampled from C_{train}. Benjamins et al. [2023] followed this protocol to study the generalizability of various model-free agents. In particular, they contrasted providing direct access to the context (as part of the observation of the agent) with simply learning on a distribution (similar to domain randomization [Tobin et al., 2017, Peng et al., 2018]) without having direct access to the context. Their findings showed that direct access to context is not always beneficial and how best to incorporate contextual information into an agent remains an open question. However, in particular, Benjamins et al. [2023] did not contrast this with agents that learn to infer context. In contrast, here, we study ZSG for agents that first need to infer their environment context and contrast this with both agents that have no access to the context and those that can directly observe it.

4 Method

We begin this section by discussing how past experiences can be used to infer the context of the environment. Further, we discuss the advantages and disadvantages of this style of inferring context in the few-shot and zero-shot settings, respectively, before using these insights to derive methodology that is particularly suited to the zero-shot setting. Finally, we outline the proposed learning algorithm that follows this methodology.

4.1 Inferring Context From Past Experiences



(a) Training of the context encoder ψ (b) Training of the policy networks π_θ, Q_ϕ.

Figure 1: Two-phase training of predictive context encoding methods. Typically, no gradient updates go through the frozen context encoder while learning the policy as depicted in (b).

The task of learning to infer the dynamics of the environment is generally done by looking at past experience. This typically involves training a context encoder ψ against an auxiliary training objective [see, e.g., Zhou et al., 2019, Evans et al., 2022]. Given a list L of transitions collected in an environment, the context encoder ψ aims to generate a latent representation l that captures relevant contextual information. Predominantly, training of ψ involves learning to predict the one-step dynamics of the environment from past trajectories based on the encoding learned in ψ. The goal of this style of learning context embedding is to directly capture how the environment evolves with each action taken.

Concretely, consider a list of observed transitions $L_c = [(s_j, a_j, s'_j) \mid j \in [0, h-1]]$ collected within a specific instance of the environment given context c, where s_j represents the state, a_j denotes the action and s'_j the subsequent state in transition j. The encoder ψ uses these transitions to learn a lower-dimensional representation l_c. The latent representation is then used by the dynamics predictor D together with the current state-action pair (s, a) and tasked with predicting the next state s'. The prediction error of D is then propagated back through ψ to ensure that ψ captures the information relevant to predict the one-step dynamics. Schematically, this can be seen in Figure 1a. Once the context encoder ψ is fully trained, the learned embedding is fixed and used to produce contextual information during subsequent training of the policy, see Figure 1b.

Inferring dynamics from previous observations has been shown to be particularly useful for few-shot adaptation in meta-RL [see, e.g., Rakelly et al., 2019, Nagabandi et al., 2019, Melo, 2022][1]. In this setting, the (learned) RL agent is allowed to update its own policy based on new observations (i.e., few shots) in the new environment. Thus, a context that aims to capture the entire transition dynamics will be helpful in exploring novel behaviors that might traverse drastically different parts of the state space. However, we argue that inferring context for zero-shot adaptation of model-free agents should not consider a learned context based on the full transition dynamics, but rather one that only captures that information, which is particularly relevant to the learned policy, which will remain unchanged at test time.

4.2 The Case for Behavior-Specific Context for Zero-Shot Generalization

Consider the example of a policy that needs to control a 4-legged robot to navigate through different worlds with varying properties, such as friction levels or gravity. At test time, in the few-shot setting, the robot might explore the landscape of the current world. Any observations made during these initial steps can be used to refine its behavior policy (e.g., change the gait to better suit the environment), and the task can be repeated as often as the particular few-shot setting allows. However, in the zero-shot setting, at test time, there is only one chance to get it right. The policy needs to be spot on from the get-go, as there is no further fine-tuning or other refinement of the policy. Thus, the context serves very different purposes in both scenarios.

The question now arises: How can we get a context that is maximally helpful to the policy at hand? Ideally, such a context should capture all the intricacies that the policy might encounter when interacting with the environment. Essentially, we want a context that is conditioned on the policy at hand. At first glance, this leads to a chicken-and-egg problem in which we need to learn a policy-conditioned context so that we can learn a context-conditioned policy or vice versa. Instead, to avoid this issue, we can use the general learning dynamics in RL that typically follows the principle of generalized policy iteration [GPI; Sutton and Barto, 2018].

In GPI we iterates between two stages (I) policy evaluation and (II) policy iteration. Starting from a random policy, we evaluate its performance. Using the evaluation data, we then improve our policy, etc. Thus, as we traverse the policy space, we get many samples of different behavior policies. These samples can be used directly to learn the latent context representation from observed past experiences in a style similar to that described in Section 4.1. To condition the context encoder on the policy behavior, we back-propagate losses of the policy objective through the context encoder. Thus, at test time, our context will capture all the information that is relevant to the current behavior of the policy. Figure 2 outlines our proposed joint context and behavior learning scheme.

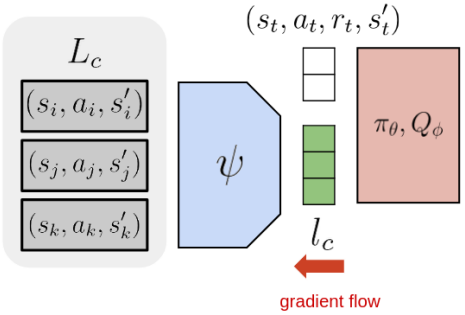


Figure 2: Joint training of the context encoder ψ and policy/value networks π_θ/Q_ϕ

4.3 Joint Context and Policy Learning

We present our proposed learning approach Algorithm 1. The pseudocode is written from the perspective of actor-critic style learning, such as the soft actor critic algorithm [SAC; Haarnoja et al.,

[1]For a recent survey on meta-RL, we refer to [Beukman et al., 2023]

2018], an off-policy deep reinforcement learning algorithm that simultaneously learns a stochastic policy π_θ and a state-action-value function Q_ϕ. However, it is important to note that our method is adaptable and can be integrated with any off-policy algorithm.

Lines 4 to 13 detail how an agent can infer its context and act with respect to it. We first sample a list of h past transitions to predict the latent context $l_c = \psi(L_c)$. We then append this learned context to the current state observation s_t, allowing us to condition the policy on both state and context. When training the policy (lines 16 to 22), we first infer the context from past transitions in a similar way. When updating the context encoder ψ, we propagate the error through the actor to condition the context on the policy.

During training, transitions are uniformly sampled from the entire replay buffer to ensure diverse experiences for the context encoder. However, at inference time, we assume the agent only has access to the transitions from the current episode (see Algorithm 2).

5 Experiments

5.1 Environments

We evaluate our context encoding method across multiple environments as described below, using the CARL library [Benjamins et al., 2023], which allows adjustments to the dynamics parameters during and between episodes. Here, we provide a brief description of the environments and contexts. For the exact training and evaluation context sets, see Table 6 in the Appendix.

- **Cartpole:** An agent needs to learn to balance a pole vertically on a moving cart which it controls. The observation space is a 4-dimensional vector (position, velocity of the cart, angle, angular velocity of the pole). We evaluate generalizability to changes in time elapsed between states.
- **Pendulum:** An agent has to learn to swing up an inverted pendulum and stabilize it at the top from a random initial position. The action space controls force direction and magnitude. We assess robustness to changes in the pendulum's length.
- **MountainCar:** The agent has to learn to drive a car up a steep slope, potentially needing to gain momentum using the opposite slope. We test adaptability to changes in power, both positive and negative values are tested. In our setting, the car is given various levels of power, allowing much faster goal reaches, potentially allowing for positive rewards.
- **Ant:** The agent has to learn to control a four-legged robot to facilitate walking. The observation space includes joint angles, velocities, contact forces, and torques (27 dimensions). We evaluate against changes in the robots' torso mass.

5.2 Baseline methods

To assess the generalization ability of our method, we compare it against the following baselines:

- **Hidden context:** No context is explicitly provided to the agent, but it is still trained across all contexts in the training set. This can be seen as similar to domain randomization [Tobin et al., 2017]. This baseline evaluates whether context-aware approaches could use context effectively when learning general policies.
- **Explicit identification:** The explicit context value c is concatenated to the observed state at each time step, at training and evaluation time. In the contextual RL setting, this is the most widely used way of learning with access to context information [Benjamins et al., 2023]. As the policy has access to complete information about the environment dynamics, i.e., the ground truth context, this baseline evaluates weather learned contexts provide benefit to the learning agents.
- **Predictive identification:** In this method, a context encoder is trained on the transition dynamics prediction task described in Section 4.2, following the training pipeline from Evans et al. [2022]. This baseline allows us to evaluate the effectiveness of jointly learning context and policy compared to learning them separately.

All baseline learning methods and our proposed method for learning with and from context are evaluated with a soft actor-critic [SAC; Haarnoja et al., 2018] agent. We keep the SAC hyperparameters

Table 1: Learning progression across *training* environments expressed as the area under the reward curve. Our method consistently achieves high AUC for all environments.

	Episodic return - area under the curve (average over 10 seeds)			
Environment	Hidden context	Explicit context	Predictive identification	Joint context & policy learning
Cartpole	**287 170**	283 847	285 018	285 528
Pendulum	−115 649	−80 745	−108 771	**−78 892**
MountainCar	1 722	**1 104 599**	103 512	510 677
Ant	57 552	−20 673	53 651	**62 282**

fixed for all methods (see Table 4 in the Appendix). We report the hyperparameters relevant for learning to infer contexts in Table 5 in the Appendix.

5.3 Measures of generalization

For each environment and method, we perform 10 independent training runs (using different random seeds) and then evaluate the trained agent for 20 episodes, resulting in a total number of $N = 200$ evaluations for each value of C_{eval}. This produces a score matrix $(s_{c_i,n})$, with $c_i \in C_{eval}$ and $n \in [1, N]$, per environment and per method. To compare the scores between different context values, we follow the methodology described by Agarwal et al. [2021] and normalize the scores by linearly rescaling them based on two reference points. We note the score of a random agent as $s_{c_i}^{random}$ and the score of an agent trained only on the default value of the context as $s_{c_i}^{default}$, and calculate the normalized scores $\bar{s}_{c_i,n}$ as:

$$\bar{s}_{c_i,n} = \frac{s_{c_i,n} - s_{c_i}^{random}}{s_{c_i}^{default} - s_{c_i}^{random}}$$

We compute the **interquartile mean (IQM)** of the agent's performance on both the interpolation and extrapolation subsets of C_{eval}, together with their respective stratified bootstrap confidence intervals, as outlined by Colas et al. [2019].

5.4 Research questions

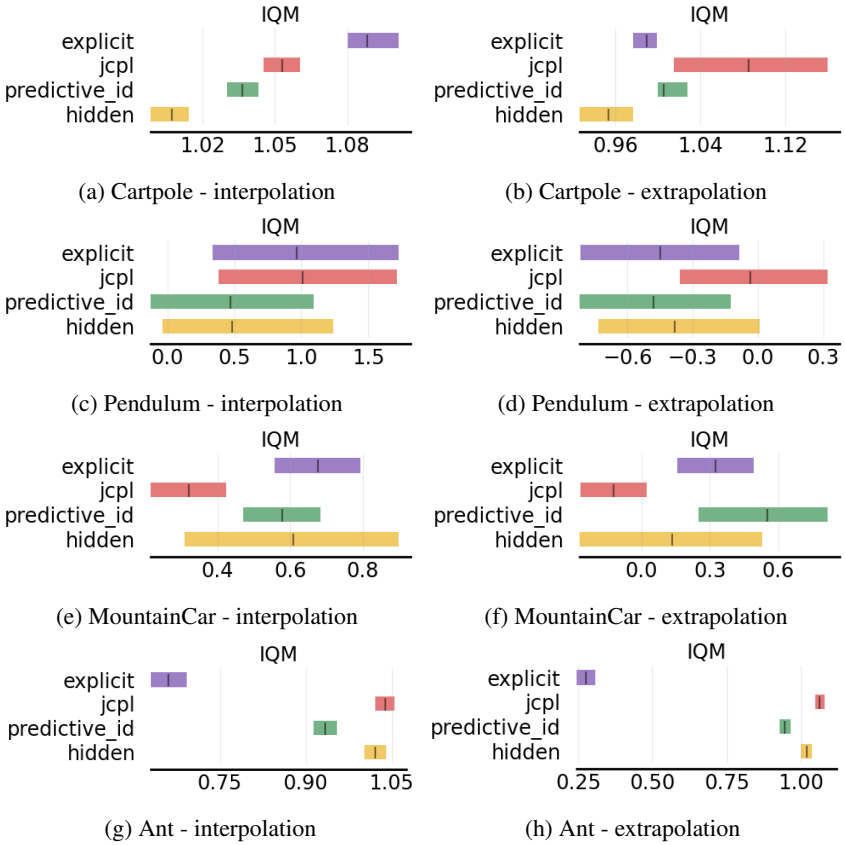
Research Question 1: Are behavior-specific context embeddings beneficial for zero-shot out-of-distribution generalization?

We evaluated the trained agents on the C_{eval} sets, and provide the IQMs of each method in Table 2. For an idea of the distribution of the scores, we also show the confidence intervals of the IQMs in Figure 3. In Cartpole and Pendulum environments, our joint context and policy learning (jcpl) method achieves higher IQM values compared to the predictive identification baseline across all settings, including interpolation, extrapolation and considering all context values.

In particular, for the Ant environment, which is relatively more complex, the results strongly favor our jcpl method, with significantly higher IQM values compared to the predictive identification baseline. Interestingly, in this environment, the explicit context baseline, which directly provides the context value as input, leads to substantially worse performance than both the jcpl and the predictive identification method, indicating that using the observed mass value directly is not beneficial to the SAC agent. This corroborates the findings of Benjamins et al. [2023]: direct access to context is not always beneficial, and how best to incorporate contextual information into an agent remains an open question. Our approach of jointly learning behavior-specific context embeddings directly addresses this open question and demonstrates improved generalization performance, especially in the complex Ant environment.

To provide further clarity for the reader, we depict the full empirical distribution of the score of each method in the Appendix (Figures 7 and 8), in the form of **performance profiles**.

Research Question 2: Do the learned embeddings capture the underlying ground truth change in the transition dynamics?



(a) Cartpole - interpolation

(b) Cartpole - extrapolation

(c) Pendulum - interpolation

(d) Pendulum - extrapolation

(e) MountainCar - interpolation

(f) MountainCar - extrapolation

(g) Ant - interpolation

(h) Ant - extrapolation

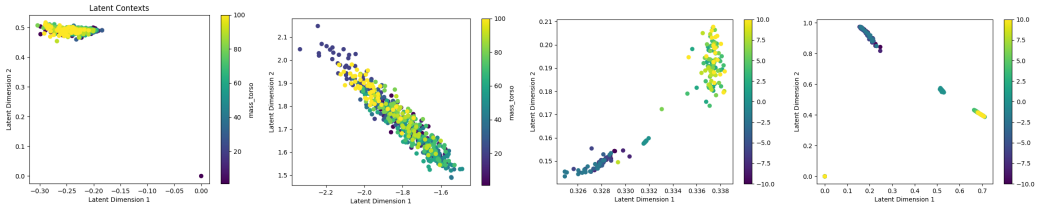
Figure 3: Interquartile Mean (IQM) of the aggregated normalized scores, along with their respective stratified bootstrap confidence intervals, in the interpolation and extrapolation settings

Table 2: Generalization metrics for the normalized scores of the jcpl and predictive identification methods. In every environmnent except MountainCar, our method scores a higher IQM, both in the Interpolation and extrapolation settings.

Environment	Metric	Interpolation	Extrapolation	All values
Cartpole	IQM jcpl	**1.052939**	**1.085407**	**1.029779**
	IQM predictive id	1.036612	1.005375	1.017251
Pendulum	IQM jcpl	**1.012740**	**-0.034381**	**0.384468**
	IQM predictive id	0.467899	-0.478549	-0.104004
MountainCar	IQM jcpl	0.320692	-0.123045	0.091731
	IQM predictive id	**0.578874**	**0.553004**	**0.588501**
Ant	IQM jcpl	**1.038206**	**1.063549**	**1.051768**
	IQM predictive id	0.934375	0.946111	0.940794

To evaluate whether the learned context embeddings effectively capture the true underlying changes in the dynamics of the environment, we examine the 2D latent representations visualized in Figure 4. In Ant and MountainCar environments, the latent embeddings learned by our jcpl method exhibit better separation between different context values compared to the predictive identification baseline. This separation suggests that the jcpl embeddings encode information about the varying transition dynamics more distinctively. Further visualizations in the Appendix, across multiple environments and training seeds, reinforce this observation.

Quantitatively, we measure the mean squared error (MSE) of a random forest model when predicting the context value from the learned latent embeddings, using 5-fold cross-validation averaged over 10 training seeds (Table 3). In the Cartpole, MountainCar, and Ant environments, jcpl achieves a lower



(a) Ant predictive identification (b) Ant jcpl (c) MountainCar predictive identification (d) MountainCar jcpl

Figure 4: Learned embeddings of both context encoding methods.

MSE compared to the predictive identification method, indicating that the learned embeddings better represent the ground-truth context information. Specifically, jcpl reduces MSE from 24.89 to 15.86 in MountainCar and from 1221.84 to 1107.84 in the complex Ant environment.

However, in the Pendulum environment, the predictive identification method achieves a slightly lower MSE of 0.0006 compared to jcpl's MSE of 0.0008. This suggests that learning the context separately may be more effective in certain environments for capturing the transition dynamics.

Overall, the qualitative and quantitative analysis demonstrate that our joint context and policy learning approach generally leads to learned embeddings that better capture the underlying ground-truth changes in the environment dynamics, particularly in more complex environments like Ant and MountainCar. By jointly optimizing context and policy representations, jcpl can discover latent embeddings that encode relevant information about the varying transition dynamics, facilitating improved generalization.

6 Conclusion

In this paper, we introduce a novel approach that seamlessly integrates the learning of context-aware policies with the ability to infer contextual embeddings, addressing the critical challenge of zero-shot generalization in reinforcement learning. By enabling the joint optimization of policy and context representations, our algorithm acquires behavior-specific embeddings that significantly enhance the adaptability of RL systems to diverse environments, eliminating the need for retraining. This unified framework represents a substantial step towards creating more autonomous and versatile RL agents.

It should be noted that the experiments primarily focused on generalization across varying context values, without explicitly considering other factors such as task variations or changes in the reward structure. Consequently, the generalization capabilities of the method in scenarios involving such variations remain an open question that warrants further investigation. Future research should focus on improving the proposed method's generalization to varied tasks by incorporating reward signals into context modeling. This can be achieved by capturing task-specific information within the learned embeddings, enabling adaptation across diverse tasks. Observing reward signals during evaluation can further enhance context identification and generalization performance, expanding the method's applicability to a wider range of tasks with variations.

Table 3: Mean square error of random forest prediction of c from l_c (5-fold cross validation, mean value across 10 training seeds)

	Mean square error	
Environment	Predictive identification	Joint context & policy learning
Cartpole	0.0013	**0.0003**
Pendulum	**0.0006**	0.0008
MountainCar	24.89	**15.86**
Ant	1221.84	**1107.84**

The current context encoder architecture averages latent context across transitions to produce a single embedding. However, exploring more advanced architectures that capture the evolution of transitions over time could yield richer and more informative context representations. By assessing the agent's uncertainty about its environment, these architectures could enhance adaptability and performance across diverse environments. This advancement in context-aware reinforcement learning could lead to more robust and versatile autonomous agents capable of operating effectively in dynamic, real-world scenarios.

References

Proceedings of the International Conference on Learning Representations (ICLR'19), 2019. Published online: iclr.cc.

S. Adriaensen, A. Biedenkapp, G. Shala, N. Awad, T. Eimer, M. Lindauer, and F. Hutter. Automated dynamic algorithm configuration. *Journal of Artificial Intelligence Research (JAIR)*, 75:1633–1699, 2022.

R. Agarwal, M. Schwarzer, P. Samuel Castro, A. C. Courville, and M. G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. Liang, J. Vaughan, and Y. Dauphin, editors, *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*. Curran Associates, 2021.

M. Andrychowicz, A. Raichuk, P. Stańczyk, M. Orsini, S. Girgin, R.ël Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski, S. Gelly, and O. Bachem. What matters for on-policy deep actor-critic methods? a large-scale study. In *Proceedings of the International Conference on Learning Representations (ICLR'21)*, 2021. Published online: iclr.cc.

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

C. Benjamins, T. Eimer, F. Schubert, A. Mohan, S. Döhler, A. Biedenkapp, B. Rosenhan, F. Hutter, and M. Lindauer. Contextualize me – the case for context in reinforcement learning. *Transactions on Machine Learning Research*, 2023.

M. Beukman, D. Jarvis, R. Klein, S. James, and B. Rosman. Dynamics generalisation in reinforcement learning via adaptive context-aware policies. In *Proceedings of the 37th International Conference on Advances in Neural Information Processing Systems (NeurIPS'23)*. Curran Associates, 2023.

A. Biedenkapp, H. F. Bozkurt, T. Eimer, F. Hutter, and M. Lindauer. Dynamic algorithm configuration: Foundation of a new meta-algorithmic framework. In J. Lang, G. De Giacomo, B. Dilkina, and M. Milano, editors, *Proceedings of the Twenty-fourth European Conference on Artificial Intelligence (ECAI'20)*, pages 427–434, June 2020.

C. Colas, O. Sigaud, and P. Oudeyer. A hitchhiker's guide to statistical comparisons of reinforcement learning algorithms. *RML@ICLR*, 2019.

Y. Duan, J. Schulman, X. Chen, P. Bartlett, I. Sutskever, and P. Abbeel. RL$ˆ2$: Fast reinforcement learning via slow reinforcement learning. *arXiv:1611.02779 [cs.AI]*, 2016.

A. Escontrela, G. Yu, P. Xu, A. Iscen, and J. Tan. Zero-shot terrain generalization for visual locomotion policies. *arXiv:2011.05513 [cs.RO]*, 2020.

B. Evans, A. Thankaraj, and L. Pinto. Context is everything: Implicit identification for dynamics adaptation. In *International Conference on Robotics and Automation, (ICRA'22)*, pages 2642–2648. IEEE, 2022.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In D. Precup and Y. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning (ICML'17)*, volume 70, pages 1126–1135. Proceedings of Machine Learning Research, 2017.

J. Guo, M. Gong, and D. Tao. A relational intervention approach for unsupervised dynamics generalization in model-based reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR'22)*, 2022. Published online: iclr.cc.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, volume 80. Proceedings of Machine Learning Research, 2018.

A. Hallak, D. Di Castro, and S. Mannor. Contextual markov decision processes. *arXiv:1502.02259 [stat.ML]*, 2015.

P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters. In S. McIlraith and K. Weinberger, editors, *Proceedings of the Thirty-Second Conference on Artificial Intelligence (AAAI'18)*. AAAI Press, 2018.

J. Kaddour, S. Sæmundsson, and M. P. Deisenroth. Probabilistic active meta-learning. In Larochelle et al. [2020].

R. Kirk, A. Zhang, E. Grefenstette, and T. Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research (JAIR)*, 76:201–264, 2023.

R. Koppejan and S. Whiteson. Neuroevolutionary reinforcement learning for generalized helicopter control. In G. Raidl et al, editor, *Proceedings of the 11th Genetic and Evolutionary Computation Conference (GECCO'09)*. Morgan Kaufmann Publishers, 2009.

H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, and H. Lin, editors. *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*, 2020. Curran Associates.

K. Lee, Y. Seo, S. Lee, H. Lee, and J. Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In H. Daume III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, volume 98, pages 5757–5766. Proceedings of Machine Learning Research, 2020.

L. C. Melo. Transformers are meta-reinforcement learners. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning (ICML'22)*, volume 162 of *Proceedings of Machine Learning Research*, pages 15340–15359. PMLR, 2022.

A. Modi, N. Jiang, S. Singh, and A. Tewari. Markov decision processes with continuous side information. In *Algorithmic Learning Theory (ALT'18)*, volume 83, pages 597–618, 2018.

A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR'19)* icl [2019]. Published online: `iclr.cc`.

J. Parker-Holder, R. Rajan, X. Song, A. Biedenkapp, Y. Miao, T. Eimer, B. Zhang, V. Nguyen, R. Calandra, A. Faust, F. Hutter, and M. Lindauer. Automated reinforcement learning (AutoRL): A survey and open problems. *Journal of Artificial Intelligence Research (JAIR)*, 74:517–568, 2022.

X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *International Conference on Robotics and Automation, (ICRA'18)*, pages 1–8. IEEE, 2018.

K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, volume 97, pages 5331–5340. Proceedings of Machine Learning Research, 2019.

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

Y. Seo, K. Lee, I. C. Gilaberte, T. Kurutach, J. Shin, and P. Abbeel. Trajectory-wise multiple choice learning for dynamics generalization in reinforcement learning. In Larochelle et al. [2020].

S. Sodhani, F. Meier, J. Pineau, and A. Zhang. Block contextual mdps for continual learning. In R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, and M. J. Kochenderfer, editors, *Learning for Dynamics and Control Conference, (L4DC'22)*, volume 168 of *Proceedings of Machine Learning Research*, pages 608–623. PMLR, 2022.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. Adaptive computation and machine learning. MIT Press, 2 edition, 2018.

J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *International Conference on Intelligent Robots and Systems (IROS'17)*, pages 23–30, 2017.

J. Wang, Z. Kurth-Nelson, H. Soyer, J. Leibo, D. Tirumala, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick. Learning to reinforcement learn. In G. Gunzelmann, A. Howes, T. Tenbrink, and E. Davelaar, editors, *Proceedings of the 39th Annual Meeting of the Cognitive Science Society*. cognitivesciencesociety.org, 2017.

L. Wen, S. Zhang, H. E. Tseng, and H. Peng. Dream to adapt: Meta reinforcement learning by latent context imagination and MDP imagination. *arXiv:2311.06673 [cs.LG]*, 2023.

W. Zhou, L. Pinto, and A. Gupta. Environment probing interaction policies. In *Proceedings of the International Conference on Learning Representations (ICLR'19)* icl [2019]. Published online: `iclr.cc`.

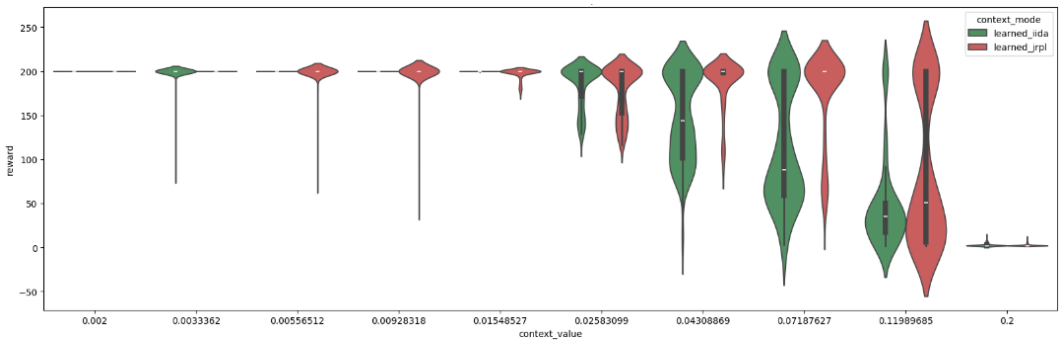
Appendix

Hyperparameter	**Value**
Total number of training steps T_{train}	3×10^4
Replay Buffer Size	10^6
Discount Factor γ	0.99
Polyak Averaging Factor τ	0.005
Batch Size b	256
Time steps before training starts	5×10^3
Learning Rate (Actor)	3×10^{-4}
Learning Rate (Critic)	10^{-3}
Optimizer (Actor, Critic)	Adam
Policy training interval	2
Target Update Interval	1
Target Entropy α_{target}	-1

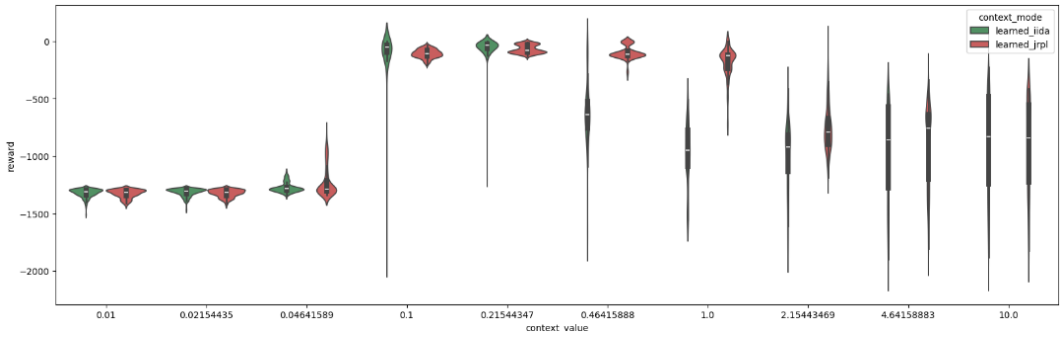
Table 4: Hyperparameters relative to the task policy algorithm (Soft Actor-Critic)

Hyperparameter	**Value**
Size h of transitions list L	20
Dimension of latent context l_c	2
Hidden dimensions (Encoder ψ, Predictor D)	[8,4]
Learning rate (Encoder ψ, Predictor D)	10^{-3}
Optimizer (Encoder ψ, Predictor D)	Adam

Table 5: Hyperparameters relative to the encoding of the context

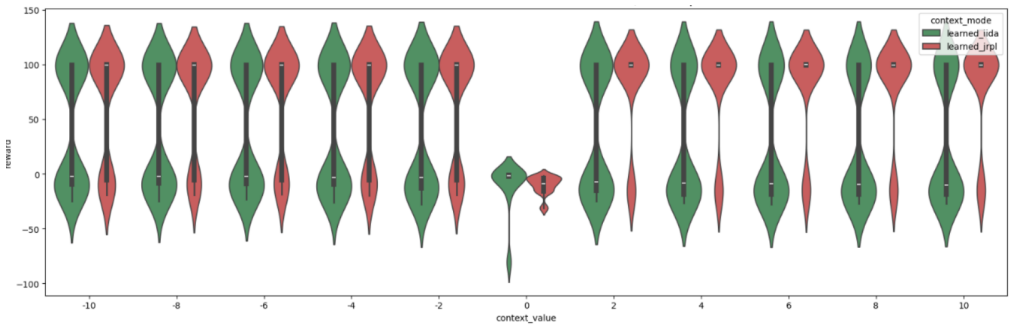


(a) Cartpole

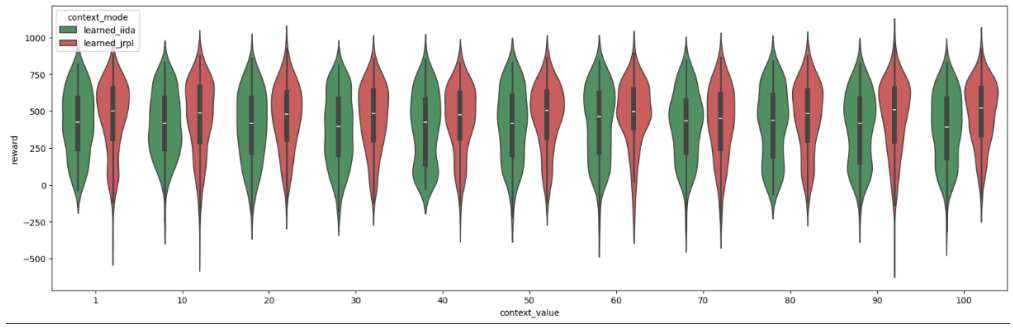


(b) Pendulum

Figure 5: Violin plots of the (non-normalized) scores for every value of C_{eval}, in the Cartpole and Pendulum environments. Our joint learning method (jcpl) is capable of zero-shot generalization to out-of-distribution environment dynamics.

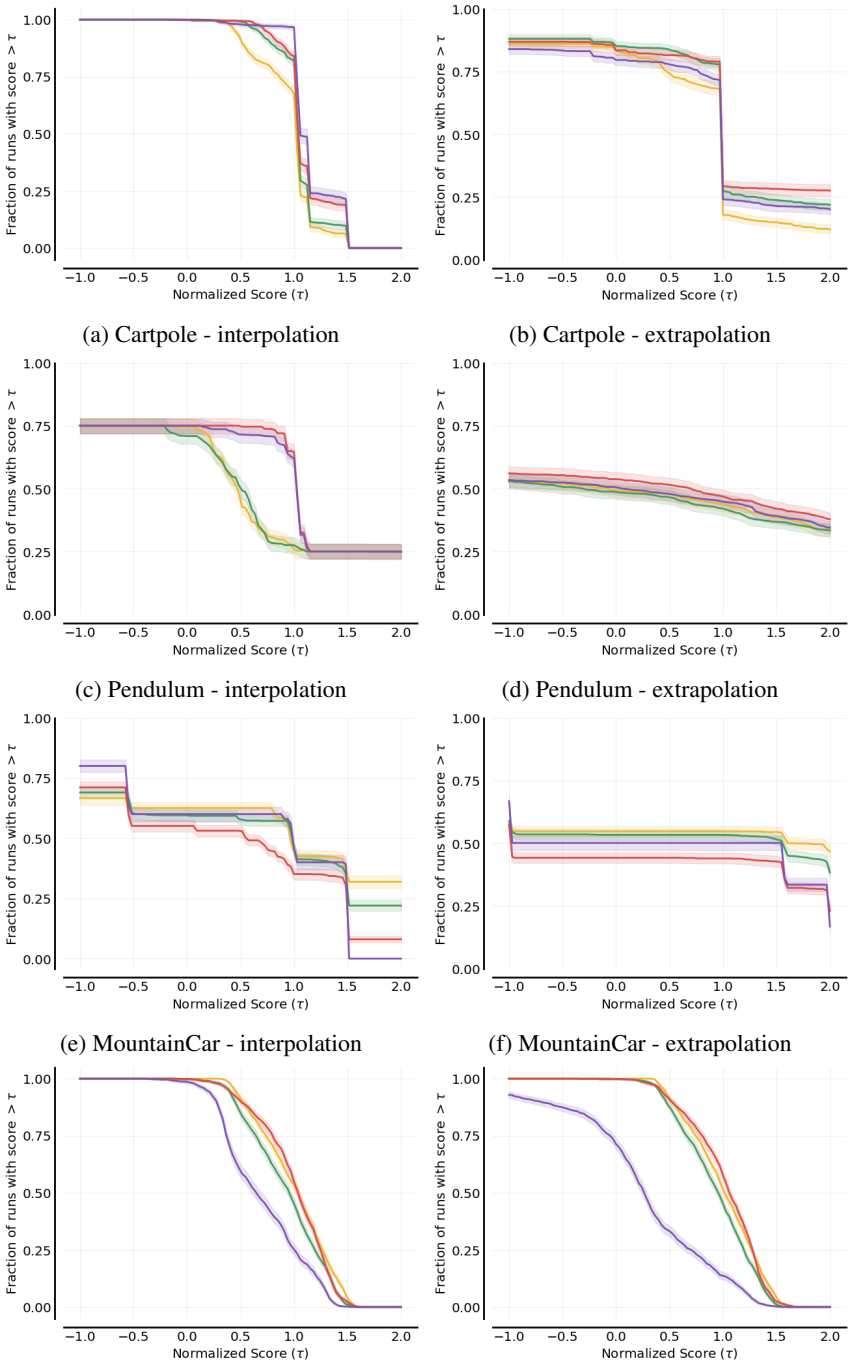


(a) MountainCar



(b) Ant

Figure 6: Violin plots of the (non-normalized) scores for every value of C_{eval}, in the MountainCar and Ant environments.



(a) Cartpole - interpolation

(b) Cartpole - extrapolation

(c) Pendulum - interpolation

(d) Pendulum - extrapolation

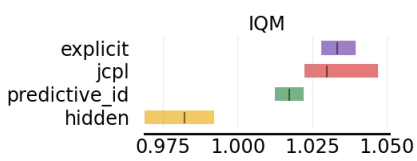
(e) MountainCar - interpolation

(f) MountainCar - extrapolation

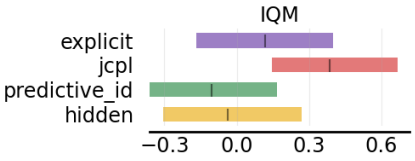
(g) Ant - interpolation

(h) Ant - extrapolation

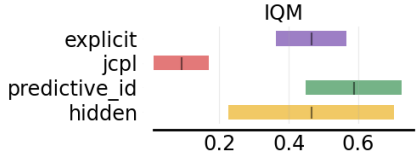
Figure 7: Performance profiles of aggregated normalized scores, in the interpolation and extrapolation settings



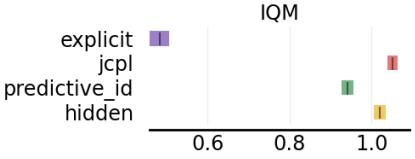
(a) Cartpole - IQM of aggregated normalized scores



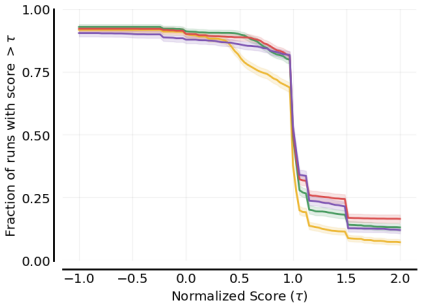
(b) Pendulum - IQM of aggregated normalized scores



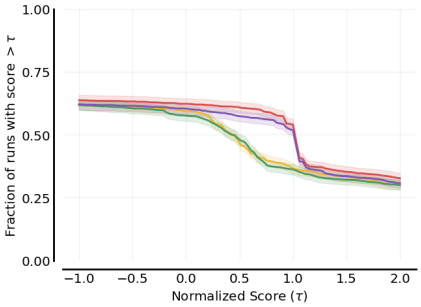
(c) MountainCar - IQM of aggregated normalized scores



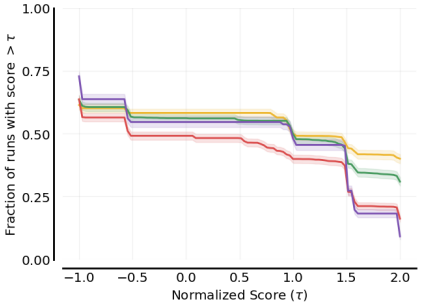
(d) Ant - IQM of aggregated normalized scores



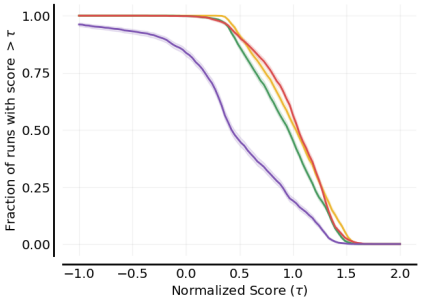
(e) Cartpole - performance profiles of aggregated normalized scores



(f) Pendulum - performance profiles of aggregated normalized scores

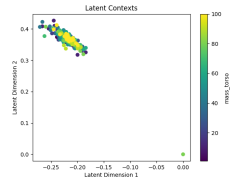


(g) MountainCar - performance profiles of aggregated normalized scores

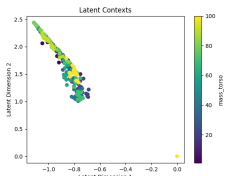


(h) Ant - performance profiles of aggregated normalized scores

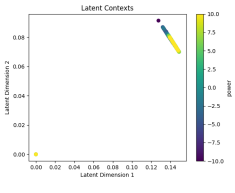
Figure 8: Interquartile Mean (IQM) and performance profiles of aggregated normalized scores, on the entire C_{eval} set



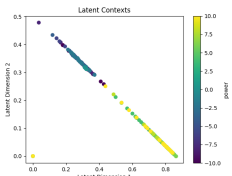
(a) Ant - predictive identification



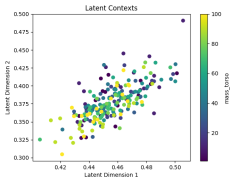
(b) Ant - jcpl



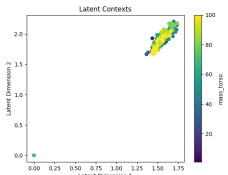
(c) MountainCar - predictive identification



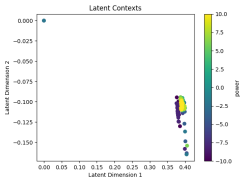
(d) MountainCar - jcpl



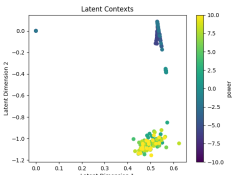
(e) Ant - predictive identification



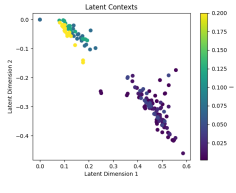
(f) Ant - jcpl



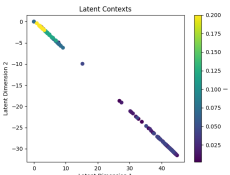
(g) MountainCar - predictive identification



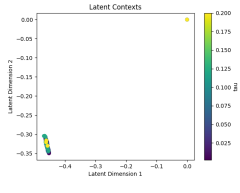
(h) MountainCar - jcpl



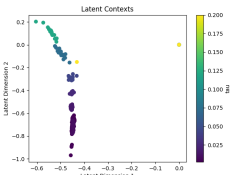
(i) Pendulum - predictive identification



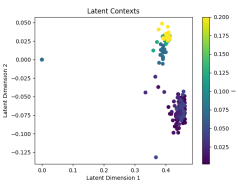
(j) Pendulum - jcpl



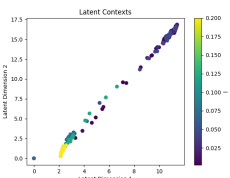
(k) Cartpole - predictive identification



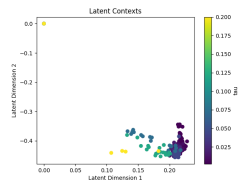
(l) Cartpole - jcpl



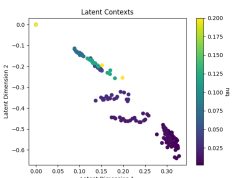
(m) Pendulum - predictive identification



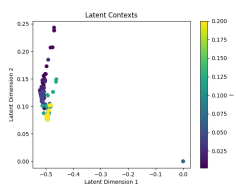
(n) Pendulum - jcpl



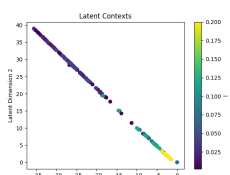
(o) Cartpole - predictive identification



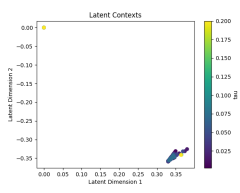
(p) Cartpole - jcpl



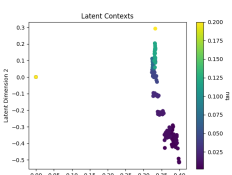
(q) Pendulum - predictive identification



(r) Pendulum - jcpl



(s) Cartpole - predictive identification



(t) Cartpole - jcpl

Figure 9: Learned embeddings of both context encoding methods, across multiple seeds

Environment ($T_{episode}$)	**Context**	C_{train}	C_{eval}
Cartpole (200)	tau	[0.007, 0.012, 0.021, 0.034, 0.057]	[0.002, 0.003, 0.006, 0.009, 0.015, 0.026, 0.043, 0.072, 0.120, 0.200]
Pendulum (200)	length	[0.07, 0.16, 0.34, 0.73, 1.58]	[0.01, 0.02, 0.05, 0.1, 0.22, 0.46, 1.0, 2.15, 4.64, 10.00]
MountainCar (999)	power	[-5, -3, -1, 1, 3, 5]	[-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10]
Ant (1000)	mass_torso	[25, 35, 45, 55, 65, 75]	[1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

Table 6: Environment settings

Algorithm 1 Training loop

Initial state s_0, context value c, context encoder ψ, actor and critic networks π_θ, Q_ϕ, number of training steps T_{train}, transition list size h, batch size b, replay buffer $R = \emptyset$
for $t \in [0, T_{train} - 1]$ **do**:

    Action Sampling
    $L_c \leftarrow Sample(\{(s_j, a_j, s'_j) \mid (s_j, a_j, r_j, s'_j, c_j) \in R, c_j = c\}, size = h)$ ▷ Sample a list L_c of transitions from context c
    if $L_c \neq \emptyset$ **then**
        $l_c \leftarrow \psi(L_c)$ ▷ Infer the context latent l_c
    else
        $l_c \leftarrow 0$
    end if
    $a_t \sim \pi_\theta(\cdot|(s_t, l_c))$ ▷ Sample action a_t from policy π_θ based on the augmented state (s_t, l_c)
    $s'_t \sim \mathcal{T}_c(\cdot|(s_t, a_t)), r_t \sim \mathcal{R}_c(\cdot|(s_t, a_t))$ ▷ Execute action a_t and observe next state s'_t, reward r_t
    $R \leftarrow R \cup (s_t, a_t, r_t, s'_t, c)$ ▷ Store transition in replay buffer

    Policy update
    $B \leftarrow Sample(R, size = b)$ ▷ Sample a mini-batch of transitions from the replay buffer
    for $(s_i, a_i, r_i, s'_i, c_i) \in B$ **do**:
        $L_{c_i} \leftarrow Sample(\{(s_j, a_j, s'_j) \mid (s_j, a_j, r_j, s'_j, c_j) \in R, c_j = c_i\})$ ▷ Sample a list L_{c_i} of transitions from context c_i
        $l_{c_i} \leftarrow \psi(L_{c_i})$ ▷ Infer the context latent l_{c_i}
    end for
    Compute the loss of the actor and critic networks using the mini-batch of augmented transitions $\{((s_i, l_{c_i}), a_i, r_i, (s'_i, l_{c_i})) \mid (s_i, a_i, r_i, s'_i, c_i) \in B\}$
    Update the context encoder ψ, actor π_θ and critic Q_ϕ networks
end for

Algorithm 2 Evaluation loop

Initial state s_0, context encoder ψ, actor and critic networks π_θ, Q_ϕ, number of episode steps $T_{episode}$, transition list $L = \emptyset$, latent context $l = 0$
for $t \in [0, T_{episode} - 1]$ **do**:
    if $L \neq \emptyset$ **then**
        $l \leftarrow \psi(L)$ ▷ Infer the context latent l
    end if
    $a_t \sim \pi_\theta(\cdot|(s_t, l))$ ▷ Sample action a_t from policy π_θ based on the augmented state (s_t, l)
    $s_{t+1} \sim \mathcal{T}_c(\cdot|(s_t, a_t))$ ▷ Execute action a_t and observe next state s'_t
    $L \leftarrow L \cup (s_t, a_t, s'_t)$ ▷ Store transition (s_t, a_t, s'_t) in L
end for