Dreaming of Many Worlds: Learning Contextual World Models Aids Zero-Shot Generalization

Sai Prasanna* **Karim Farid*** **Raghu Rajan** **André Biedenkapp**
{ramans, faridk, rajanr, biedenka}@cs.uni-freiburg.de
University of Freiburg

Abstract

Zero-shot generalization (ZSG) to unseen dynamics is a major challenge for creating generally capable embodied agents. To address the broader challenge, we start with the simpler setting of contextual reinforcement learning (cRL), assuming observability of the context values that parameterize the variation in the system's dynamics, such as the mass or dimensions of a robot, without making further simplifying assumptions about the observability of the Markovian state. Toward the goal of ZSG to unseen variation in context, we propose the contextual recurrent state-space model (cRSSM), which introduces changes to the world model of Dreamer (v3) [Hafner et al., 2023]. This allows the world model to incorporate context for inferring latent Markovian states from the observations and modeling the latent dynamics. Our approach is evaluated on two tasks from the CARL benchmark suite, which is tailored to study contextual RL. Our experiments show that such systematic incorporation of the context improves the ZSG of the policies trained on the "dreams" of the world model. The code for all our experiments is available at `https://github.com/sai-prasanna/dreaming_of_many_worlds`.

1 Introduction

Model-Based Reinforcement Learning (MBRL) promises to be one of the most data-efficient frameworks for learning control. With this data efficiency, MBRL could significantly impact real-world applications, such as robotics and autonomous systems, for which efficient learning and generalization are paramount. Recent MBRL approaches are capable of achieving performance comparable to model-free reinforcement learning (MFRL) algorithms while only requiring a fraction of the data [see, e.g., Chua et al., 2018, Hafner et al., 2020, 2021, 2023, Wu et al., 2022, Hansen et al., 2024].

A key challenge in MBRL is the ability to generalize to unseen environments, particularly in a *zero-shot* setting, where an agent must perform effectively in novel scenarios without prior experience. Although MBRL has shown great improvement in recent years, both MBRL and MFRL algorithms remain susceptible to small changes in environment dynamics [Kirk et al., 2023]. This can be attributed in part to the complexity of the (MB)RL pipeline [Zhang et al., 2021b, Parker-Holder et al., 2022] but also to a lack of understanding, as zero-shot generalization (ZSG) remains an understudied domain for MBRL [Kirk et al., 2023].

An influential family of MBRL algorithms is Dreamer [Hafner et al., 2020, 2021, 2023]. Dreamer-like algorithms learn a latent representation of the world from which plausible trajectories can be imagined that can be used to improve decision-making. The family of Dreamer algorithms has achieved impressive results in various domains in both learned policy performance and sample efficiency during learning. However, Dreamer-like algorithms have not yet been studied in the zero-shot generalization setting.

Here, we propose to use the contextual reinforcement learning paradigm [Kirk et al., 2023, Benjamins et al., 2023] to study Dreamers' learning capabilities within and across many worlds. To this end,

*equal contribution

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

we assume that we have access to privileged information about how the transition dynamics of the underlying Markov decision process (MDP) is parameterized, i.e., the context[Hallak et al., 2015]. We use tasks from the Contextual and Adaptive Reinforcement Learning benchmark [CARL; Benjamins et al., 2023] where the context defines some physical properties that affect the dynamics and that an RL agent can observe while trying to solve a given task. Examples of such properties are gravity or the mass of a load that a robot might lift. We assume the context parameters are continuous and hence can be studied meaningfully for interpolation and extrapolation, unlike discrete parameters where generalization to unseen values is not well-defined.

We analyze Dreamer's ZSG capabilities, in- and out-of-distribution (OOD), when naively integrating context, and we propose an improved Dreamer variant that integrates context more intelligently and demonstrates improved generalization abilities. In particular, our contributions are as follows.

- We provide the first principled study in understanding Dreamer's generalization capabilities for in- as well as out-of-distribution (OOD) tasks on two tasks from CARL; and
- We propose a novel approach for conditioning the Dreamer architecture on context and show how it improves Dreamer's zero-shot generalization ability on the given tasks.

2 Related Work

Our approach aims to improve the ZSG of MBRL agents. As such, in this section, we discuss related works from meta-RL, an area aimed at improving few- and zero-shot generalization, followed by MBRL and ZSG in MBRL.

Meta-RL Meta-reinforcement learning (meta-RL) [Beck et al., 2023] has been proposed as a promising approach to address the challenge of generalization in RL. Meta-RL aims to learn an RL agent that can adapt to new tasks in a sample-efficient manner. Meta-RL algorithms [see, e.g., Duan et al., 2016, Wang et al., 2017, Nagabandi et al., 2019, Rakelly et al., 2019, Melo, 2022, Wen et al., 2023] are designed to quickly adapt to new and unseen settings with limited access to new experiences (i.e., few-shot adaptation) generated by the RL agent. In contrast, our work focuses on zero-shot generalization (ZSG) for RL [Kirk et al., 2023], where we aim to learn policies that are capable of zero-shot adaptation to new settings without assuming access to further training or the reward signal.

Model-Based RL MBRL is believed to be one of the most promising directions to improve the sample efficiency of RL algorithms. Young et al. [2023] make the case that algorithms that use experience with a model can generalize to unseen environments better than those that rely purely on value-function generalization and experience replay. Empirically, MBRL algorithms, such as Dreamer [Hafner et al., 2020, 2023] and TD-MPC2 [Hansen et al., 2024] pipelines, achieve state-of-the-art sample efficiency. While Dreamer's focus was on achieving a high return in a variety of individual environments, our focus is to study and improve generalization capabilities in contextual variants of environments and we evaluate our work on currently available environments tailored for this. Based on Dreamer's success, we build our approach on it and study its zero-shot generalization capabilities before suggesting an improved approach. Dreamer has recently been studied in the meta-RL case for a few-shot generalization [Wen et al., 2023]. However, this work still requires many interactions with the target domain for the agent to learn to adapt to a test task successfully.

Zero-Shot Generalization in MBRL Studies on zero-shot generalization (ZSG) have mainly focused on the model-free case [Kirk et al., 2023]. The few works that have studied zero-shot generalization in MBRL assume that the context is not observable by an agent, necessitating context inference via separate latent variables with different update frequency (forming a hierarchy) or enforcing latent states to encode such contextual information to allow adaptation [Lee et al., 2020, Perez et al., 2020, Zhang et al., 2021a, Ball et al., 2021, Guo et al., 2022, Sodhani et al., 2022, Wen et al., 2023, Sekar et al., 2020, Gumbsch et al., 2024, Ying et al., 2024]. In contrast, our work is more similar to the study of Benjamins et al. [2023], which assumes that the context is observable and accessible by an agent. Benjamins et al. [2023] evaluated multiple model-free agents for ZSG. Beukman et al. [2023] build on this approach and propose to learn a hypernetwork to adapt a SAC [Haarnoja et al., 2018] agent policy based on the observed context. We believe that the observable context setting holds a lot of merit, as, on the one hand, various physical contexts

that can be sensed by a real robot, such as the mass of a load it carries, could be used to improve its policy in different contexts [Escontrela et al., 2020]. On the other hand, insights gained in this observable context setting will likely be useful for the more challenging setting where context is hidden and needs to be inferred. Furthermore, while our work assumes an observable context, it still tackles the challenging setting of partial observability in the underlying latent Markovian state. Our work contrasts previous work in ZSG that operates under the assumption that the latent state is observable or can be decoded from purely high-dimensional observations [Du et al., 2019]. To tackle such partial observability, we design a systematic method to use context to estimate latent states when the context is visible. An early work in this direction learns policies for helicopter control under partial observability [Koppejan and Whiteson, 2009].

3 Background

Zero-Shot Generalization in Contextual Markov Decision Processes To empirically study ZSG in a partially observable setting, we use a definition of the contextual MDP [cMDP; Hallak et al., 2015, Modi et al., 2018] similar to the one proposed by Kirk et al. [2023]. A cMDP is a tuple $M := (S, A, O, R, T, C, \phi, p(s_0 \mid c), p(c))$ where S is the state space, A is the action space, O is the observation space, $R\colon S \times C \times A \to \mathbb{R}$ is the reward function, $T\colon S \times A \times C \to \text{dist}(S)$ is the stochastic Markov transition function over the states. C is the context space. $\phi\colon S \times C \to O$ is the observation emission function, $p(s_0 \mid c)$ is the initial state distribution for a given context, and $p(c)$ is the context distribution with $c \in C$. For the commonly used discrete-time case, the timesteps are $t \in [0, H]$, with H as the horizon per episode. Context remains the same during an episode, but may change across episodes. For a given cMDP, we can train a policy $\pi : p(a_t \mid o_{\leq t}, a_{<t}, c)$ that is trained with the objective of maximizing the expected sum of rewards $\mathbb{E}_\pi(\sum_{t=0}^{H} r_t)$ in the distribution of the training contexts $p_{\text{train}}(c)$. We can then study the generalizability of this policy in the zero-shot setting by evaluating transfer on an evaluation context distribution $p_{\text{eval}}(c)$.

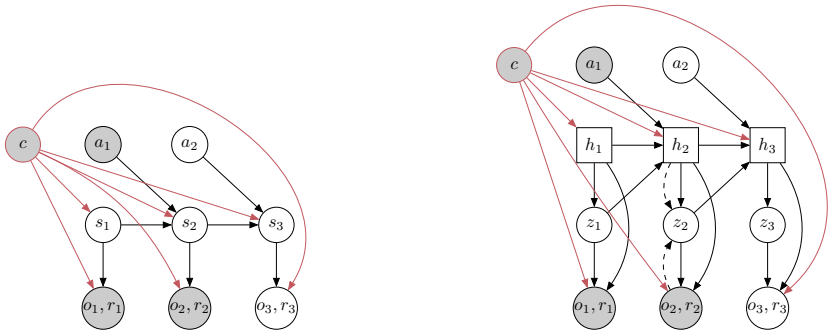
Importance of Context We often face partial observability of latent Markovian states in real-world RL tasks. Providing context to an agent may help infer such latent Markovian states. As a motivating example, consider a wheeled robot that has to deliver goods to various locations, encountering varying terrains along the way, including rough terrains that may damage the robot's wheels and, ultimately, the robot. Now, assume that the robot has a sensor to measure its velocity. If the robot must navigate without damaging itself, it may be useful to estimate the coefficient of friction of the surface on which it moves. This coefficient of friction is a non-stationary latent state variable that changes within an episode. Here, the mass of the load on the robot (context), the coefficient of friction of the terrain (latent state), and the torque applied to the wheels (action) causally affect the velocity (observation) of the robot. The robot must use its context, observations, and actions to infer the friction. The robot could then use this estimate of the friction to improve its policy and decide whether to apply more or less torque. This example, though oversimplified, gives us an idea of how effective the context might be to learn to infer latent states in partially observable settings and to use the latent state to improve its policy. It motivates the design of our contextual Dreamer agent, which we describe now.

4 Method

We now discuss how we incorporate context into the Dreamer (v3) [Hafner et al., 2023] algorithm. We first introduce our novel approach to contextual dreaming and then contrast it with naive ways of learning from and with context information.

4.1 Contextual Dreamer

We employ a novel contextual recurrent state space model (cRSSM) that builds on Dreamer's RSSM world model and systematically introduces context. Here, we discuss how it can be used to imagine trajectories in the contextual RL setting and describe how we alter Dreamer's actor-critic policy network to use the context and the latent states inferred by the contextual world model.



(a) Generative Model for a cMDP

(b) cRSSM

Figure 1: Latent dynamics models. The models shown observe the first two time steps and predict the third. Circles represent stochastic variables, and squares represent deterministic variables. Solid lines denote the generative process, and dashed lines denote the inference model. The context node and edges are highlighted in red. (a) The generative model for a cMDP. (b) Our cRSSM.

4.1.1 Contextual Recurrent State-Space Model (cRSSM)

We first define a non-linear latent space model (see Figure 1a) for the general formulation of a cMDP (see Section 3) with partial observability. This defines the generative process of observations $\{o_t\}_{t=1}^{H}$ and rewards $\{r_t\}_{t=1}^{H}$ from latent states $\{s_t\}_{t=1}^{H}$, actions $\{a_t\}_{t=1}^{H}$, and context c. This generative model describes the influence of context on the transition dynamics, rewards, and observations.

World Model Objective To perform inference of the latent states for this non-linear model, we cannot directly compute the posterior [Hafner et al., 2019]. Instead, we learn an encoder $q(s_{1:H} \mid o_{1:H}, a_{1:H}, c) = \prod_{t=1}^{H} q_\theta(s_t \mid s_{t-1}, a_{t-1}, o_t, c)$. This encoder incorporates context to estimate latent states from observations and actions. Using the encoder, we follow Hafner et al. [2019] in constructing a variational bound on the data log-likelihood. Here, we write the objective for predicting only the observations; a similar derivation applies for the rewards and the prediction of the continuation flag[2] of the episode $n_t \in 0, 1$. The evidence lower bound (ELBO) obtained using Jensen's inequality is then

$$\begin{aligned}\ln p(o_{1:T} \mid a_{1:T}, c) &\triangleq \ln \int \prod_t p(s_t \mid s_{t-1}, a_{t-1}, c) p(o_t \mid s_t, c)\, ds_{1:T} \\ &\geq \sum_{t=1}^{T} \Big(\underbrace{\mathrm{E}_{q(s_t \mid o_{\leq t}, a_{<t})}[\ln p(o_t \mid s_t, c)]}_{\text{reconstruction}} - \underbrace{\mathrm{E}_{q(s_{t-1} \mid o_{\leq t-1}, a_{<t-1}, c)} \big[\mathrm{KL}[q(s_t \mid o_{\leq t}, a_{<t}, c) \parallel p(s_t \mid s_{t-1}, a_{t-1}, c)]\big]}_{\text{complexity}} \Big).\end{aligned}$$

We mainly extend the steps in Hafner et al. 2019 for constructing the lower bound with context (derivation in Appendix A). The expectations in this objective can be optimized with gradient ascent on samples drawn from the encoder using the reparameterization trick (Kingma and Welling [2014]).

Models We follow the Dreamer (v3) algorithm's (Hafner et al. [2023]) choice to split each latent state s_t into a deterministic state h_t and a stochastic state z_t. This defines the cRSSM model (see Figure 1b), which can be split into the following models:

Deterministic state model:	$h_t = f_\theta(h_{t-1}, z_{t-1}, a_{t-1}, c)$
Stochastic state model:	$\hat{z}_t \sim p_\theta(\hat{z}_t \mid h_t)$
Encoder	$z_t \sim q_\theta(z_t \mid h_t, o_t)$
Observation model:	$\hat{o}_t \sim p_\theta(\hat{o}_t \mid h_t, z_t, c)$
Reward model:	$\hat{r} \sim p_\theta(\hat{r}_t \mid h_t, z_t, c)$
Continue model:	$\hat{n} \sim p_\theta(\hat{n}_t \mid h_t, z_t, c)$.

Refer to Appendix F for an intutive explanation of how the RSSM and cRSSM work.

[2]Continuation flag indicates whether the state is terminal.

Parameterizing the Models We do not change Dreamer's neural network architecture choices to parameterize these models. To train the objective, the Dreamer algorithm uses the past experiences of the agent (an actor-critic policy), which is trained concurrently with the cRSSM.

4.1.2 Dreaming of Many Worlds

Starting from a state s_τ, inferred at some timestep τ from an observation sequence $o_{1:\tau}$ and actions $a_{1:\tau-1}$ and the true or factual context c_F for that sequence, we can use the cRSSM to sample trajectories in the latent state space. The cRSSM also allows for imagining trajectories for counterfactual contexts, or "dreaming of many worlds". We can do so by switching the context c_F, which governs the episode where the observations ($o_{1:\tau}$) used to infer the start state of the imagination s_τ were generated and dreaming further from that point in a different counterfactual context c_{CF}.

4.1.3 Actor-Critic Policy

We largely follow Dreamer(v3) regarding training an actor-critic policy on imagined trajectories. However, we introduce context into policy learning by conditioning the actor and critic networks with the context. The actor is optimized to maximize the expected return on the imagined trajectories.

$$\text{Actor:} \quad a_\tau \sim \pi_\phi(a_\tau \mid s_\tau, c); \text{Critic:} \quad v_\psi(s_t, c) \approx \mathrm{E}_{\pi(\cdot|s_\tau,c)}[\textstyle\sum_{\tau=0}^{H-t} \gamma^\tau r_{t+\tau}].$$

4.2 Naive Use of Context in Dreamer

As discussed previously, Dreamers ZSG capabilities have not been explored in an observable context setting, nor, to the best of our knowledge, has this setting been explored in MBRL in general. Thus, we propose and discuss naive learning variants from and within the contextual setting and contrast them with our proposed cRSSM. The naive variants are then used as baselines in our experiments.

4.2.1 Context as an Observation

A commonly adopted approach to incorporate the inferred or true context into an algorithm is to concatenate it with the state or observation [Perez et al., 2020, Biedenkapp et al., 2022, Sodhani et al., 2022, Benjamins et al., 2023]. We study applying this approach to vanilla Dreamer. Vanilla Dreamer optimizes all the objectives defined in Section 4.1, but it does so without incorporating context in those objectives. To incorporate context naively, we provide it concatenated with the observation to the stochastic state encoder. $q_\theta(z_t \mid h_t, x_t)$ where $x_t \doteq [c_t, o_t]$. Note that only the encoder gets the observation (or here the observation with context) as an input. The decoder then has to learn to reconstruct the context as it is part of the observation. The latent dynamics predictor used for imagination does not condition on observations. For a consistent imagination, the RSSM is burdened with retaining the context value (provided as an observation) which got encoded into the latent state, from which imagination begins. Since we only provide the latent state inferred by the encoder to the actor-critic model in this setting, if the context is not retained, then the actor-critic network will also not have access to the context information to learn accurate policies. This could make it hard to generalize OOD. Still, the simplicity of directly using context as part of the observation is appealing, which has led to it being the predominant approach in model-free cRL.

4.2.2 Hidden Context

The cMDP is a sub-class of POMDP (Kirk et al. [2023]). As the vanilla Dreamer algorithm applies to POMDPs, we can use it without modification in the cRL setting without providing the context but training on episodes drawn from a training distribution over contexts. This is similar to the *domain randomization* [Tobin et al., 2017, Peng et al., 2018, Andrychowicz et al., 2020] where the aim is to train the policy on a context distribution, usually inside the simulator, to aid generalization to some target context, usually on the real world. While applying domain randomization, most approaches aim to cover the target context distribution upon which they aim their policies to generalize. While studying ZSG, we also care about OOD contexts.

Unlike context-unaware model-free domain randomization approaches that learn representations purely from the reward signal, Dreamer's world modeling objective provides a useful inductive bias that could allow the model's observation encoder to learn how to infer the context implicitly more efficiently. With the clear disadvantage of not using context information when it is available,

this approach might not be able to learn to distinguish which exact context setting it properly is dealing with. Consequently, the resulting policies might act for a spurious context and thus behave sub-optimally or even fail catastrophically. However, providing a training distribution of contexts might already be enough for the world model to infer the context, especially if the context is implicitly encoded in the observations (e.g., the pole length in CartPole with pixel observations). Thus, this style of context handling can be viewed as a simple context-inference approach. In this setting, similar to treating context as an observation, we only use the latent state inferred by the encoder as the input to the policy network.

5 Experiments and Discussion

In this section, we assess the performance of Dreamer in achieving generalization under observed contexts. We compare various context conditioning approaches, following the evaluation protocols outlined in [Kirk et al., 2023]. In particular, our findings highlight the effectiveness of our cRSSM method, showcasing quantitative and qualitative results in terms of zero-shot generalization (ZSG), particularly in scenarios involving changes to context parameters affecting the dynamics.

Environments: Our experiments leverage CARL, tailored for our investigation into ZSG. In CARL we pick the following environments and contexts.

- **CartPole** [Barto et al., 1983]: `pole length` and `gravity`.
- **DMC Walker Walk** [Tassa et al., 2018]: `actuator strength` and `gravity`.

As these environments are good examples of the desired continuous contextual settings, they provide a suitable benchmark for our study. CartPole serves as a simple problem, while DMC Walker presents a more complex challenge. This approach aligns with prior works, such as Sodhani et al. [2022] and Zhang et al. [2021a], which evaluate generalization on DMControl tasks to changes in context. For each environment, we use two modalities of observation, namely (1) **Featurized:** This uses featurized observations, which are generally easier to learn policies as they exhibit the least or no partial observability depending on the environment; (2) **Pixel:** Image observations which are more difficult as the model has to infer the latent states from it.

Training Pipeline We use Dreamer (v3) default hyperparameters for all experiments, with 50k steps for CartPole and 500k for DMC Walker (10 seeds). We also show DMC Walker results with 10 seeds and 100k steps in Appendix H to analyze performance with fewer samples. Refer Appendix G for exact hyperparameter values.

The CARL benchmark provides default context values (i.e., those commonly used in the literature for single-environment training) for the two context dimensions we consider for each environment. For each context type, we define *training* and *evaluation* ranges. The default value, training ranges, and evaluation context values are provided in Appendix C. We train our three methods, namely the *cRSSM*, *concat-context* and *hidden-context*, in the following *training settings*:

1. **Single context variation**: We randomly sample 100 context values uniformly for one context dimension in its training range, keeping the other fixed to its default value and vice versa.
2. **Dual context variation**: We sample 100 context values uniformly in the training range of both context dimensions.

Evaluation Protocol Following Kirk et al. [2023] we evaluate our agents on the following *evaluation settings* (visualized in appendix B):

1. **Interpolation (I)**: Evaluation contexts are selected fully within the training range.
2. **Inter+Extrapolation (I+E)**: Evaluation contexts are selected to be within the training distribution for one context dimension and out-of-distribution (OOD) for another. This evaluation setting only applies to agents trained in the dual context variation setting.
3. **Extrapolation (E)**: Evaluation contexts are fully OOD, as they are selected outside the training context set limits.

To gain insights into Dreamers' basic generalization capabilities, we also train context-unaware agents on the single default context (*default-context*) per environment. We evaluate these hidden context default agents in the same context values used for the three evaluation settings to compare each of our methods trained in the two training settings. In the evaluation protocol, each context could constitute a task of different difficulty. For example, learning to control an agent in a context where it carries lighter loads might be easier than the one with heavier loads. Following Benjamins et al. [2023], to obtain an upper bound of the policy returns, we train *expert agents* for selected contexts that broadly cover our ranges of training and evaluation contexts. Expert agent performances are the best mean return over 50 episodes among five seeds. This gives an upper bound on the returns achievable if Dreamer is trained in a particular context. We used the best *random policy* mean return on 50 episodes over five seeds to define a lower bound. Refer to Appendix D.1 for the performance of expert and random policies in each environment and context value.

Evaluation Metric We use the performance of the expert and random policies to normalize our evaluation performance. A normalized score of 1.0 would indicate the expert performance of an agent in that setting, and 0.0 performance equal to a random policy. Since we evaluate our approaches on more contexts than the number of experts, we pick the nearest context (normalized to account for different scales of contexts) for which an expert is available and use it as reference.Following the recommendation of Agarwal et al. [2021], we report the interquartile mean (IQM) of the normalized aggregated scores across contexts in different regions for each experiment.

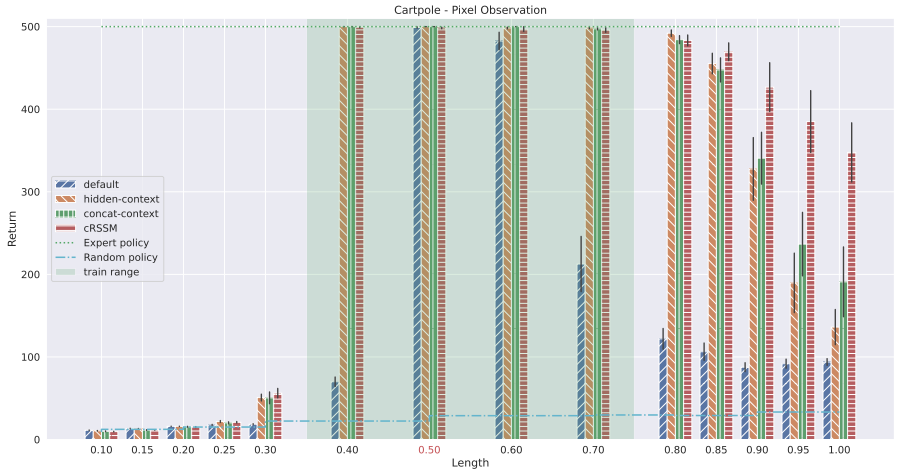
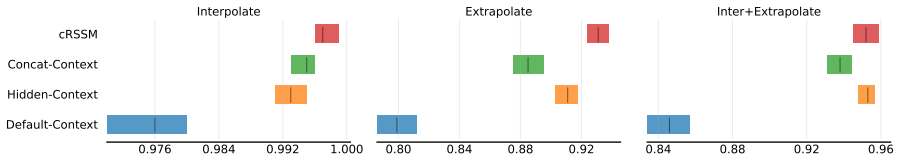


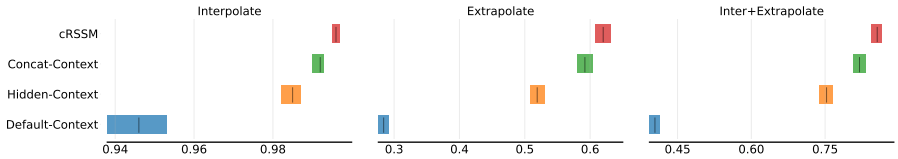
Figure 2: Generalization capabilities of Dreamer with pixel-observations when varying the pole length in CartPole. The y-axis indicates the gained reward, and the x-axis the pole length. The blue bars shows vanilla Dreamers performance when only training on the default length (0.50) and extrapolating to other settings. The shaded are gives the training range for the methods using context. Expert and random policies give upper and lower bound for performances in each context.

5.1 Results

In this section, we analyze our results to answer three key research questions that motivate our study of ZSG and our method of context conditioning. To help answer the first two questions, we first provide the results on the representative Cartpole with pixel observations setting, comparing the mean evaluation returns across our methods in Figure 2. To compare the overall performance of our different modalities for our four methods, in Figure 3, we provide the aggregated IQM over normalized return along with stratified bootstrap 95% confidence intervals Agarwal et al. [2021] across different contexts, single/dual-variation training paradigms, and environments. We also present the aggregated probability of improvement for cRSSM compared to other methods in Appendix D.4. Individual results comparing the raw returns of all agents in different contexts, modalities, and context variation settings are available in Appendix D. We also report the aggregated IQM scores for different context regions for each individual setting in Table 1. Refer Appendix D.5 plots for these individual IQMs and 95% confidence intervals.



(a) Feature based IQM



(b) Pixel based IQM

Figure 3: Aggregated comparison (across contexts & tasks) across *cRSSM*, *concat-context*, *hidden-context*, and *default-context* for the evaluation settings: Interpolation, Extrapolation, and Inter+Extrapolation using IQM over expert normalized scores for both input modalities. Intervals shown are stratified bootstrap 95% confidence intervals over seeds & aggregated contexts.

5.1.1 How Effective is Domain Randomization for Dreamer's ZSG?

To answer this, we compare the two approaches *default-context*, with the agent trained on the default context c_d, and the *hidden-context*, which involves training with domain randomization of contexts. As a motivating example to compare these methods, we first present a representative result in Figure 2 for the different methods trained on the Cartpole environment with pixel observations and varying the pole length. We observe that the *hidden-context* agent significantly outperforms the *default-context* agent, especially in the extrapolation setting. The performance of *default-context* agent drops noticeably when it moves away from its familiar default context. The aggregated results across the interpolation and extrapolation regime for this setting are available in the pixel column under the rows (l d/h/c/cR) for the Cartpole group in Table 1. The aggregated metrics in Figure 3 show that *hidden-context* outperforms *default-context* in all settings. The improvement is more pronounced in the pixel-based modality (Figure 3b). This highlights the impact of domain randomization for generalization to unseen contexts in the more complex pixel modality.

In summary, domain randomization benefits the ZSG of the Dreamer algorithm, and the improvement is striking for the pronounced pixel modality on the evaluated tasks.

5.1.2 Does Explicit Context Conditioning Aid ZSG?

Having established the benefits of domain randomization through *hidden-context* for Dreamer's ZSG on the given tasks, our focus shifts to evaluating the impact of explicit context conditioning methods, namely *cRSSM*, our principled way to incorporate context into Dreamer's world model; and *concat-context* where we augment the observations with the context.

In the Cartpole environment, for the pixel modality observations (Figure 2), both explicit conditioning methods, *cRSSM* and *concat-context*, demonstrate superior performance over *hidden-context*, particularly in scenarios with longer pole lengths. Here, *cRSSM* emerges as the frontrunner. To extend this analysis to all of our settings, we again turn to the aggregated IQM scores. For the featurized modality (Figure 3a), the *cRSSM* significantly outperforms both *hidden-context* and *concat-context*. In contrast, *concat-context* trails behind *hidden-context* in *inter+extrapolation* and extrapolation settings. In the more challenging pixel modality (refer to Figure 3b), explicit context conditioning techniques demonstrate significant improvements over the *hidden-context* across all evaluation scenarios, highlighting the importance of context conditioning for generalization.Between the explicit context conditioning methods, cRSSM performs best in all evaluation regions on aggregate. The improvements are particularly pronounced in the more challenging extrapolation and *inter+extrapolation* scenarios. Following Agarwal et al. [2021], we provide the probability of improvement of cRSSM over other methods in Appendix D.4, solidifying our claims.

For a detailed breakdown of each task, context variation, and evaluation protocol, we consult Table 1. In the Cartpole environment, in the featurized case, all methods perform similarly in all settings and

Table 1: Results for different evaluation settings, in featurized and pixel modality. Each described by three variables: context, method, and mode. Context takes values from $\{c_1, c_2\}$ where c_1 is gravity for both CartPole and Walker environments, c_2 is pole length for CartPole and Actuator Strength for Walker. c_{1+2} indicating multiple contexts; and method from $\{d$: default-context, h : hidden-context, c : concat-context, cR : cRSSM$\}$

Setting	I	E	I+E	I	E	I+E	I	E	I+E	I	E	I+E
	CartPole						Walker					
	Featurized			Pixel								
(c_1 d)	1.000	1.000	-	1.000	0.938	-	0.903	0.561	-	0.940	0.546	-
(c_1 h)	1.000	1.000	-	1.000	0.995	-	0.967	0.764	-	0.945	0.708	-
(c_1 c)	1.000	1.000	-	1.000	0.997	-	0.966	0.769	-	**0.966**	**0.733**	-
(c_1 cR)	1.000	1.000	-	1.000	**1.000**	-	**0.985**	**0.806**	-	0.959	0.710	-
(c_2 d)	1.000	0.995	-	0.677	0.059	-	0.885	0.479	-	0.959	0.461	-
(c_2 h)	1.000	0.996	-	**1.000**	0.169	-	0.959	0.571	-	0.947	0.571	-
(c_2 c)	1.000	0.987	-	**1.000**	0.210	-	0.926	0.597	-	0.983	**0.635**	-
(c_2 cR)	1.000	**1.000**	-	**1.000**	**0.374**	-	**0.998**	**0.674**	-	**0.994**	0.623	-
(c_{1+2} d)	1.000	0.945	0.998	0.901	0.038	0.210	0.842	0.520	0.595	0.915	0.503	0.570
(c_{1+2} h)	1.000	0.989	**1.000**	**1.000**	0.149	0.701	0.966	**0.764**	**0.843**	0.952	0.666	0.772
(c_{1+2} c)	1.000	0.970	**1.000**	**1.000**	0.257	0.779	0.972	0.727	0.830	0.965	**0.724**	0.823
(c_{1+2} cR)	1.000	**0.997**	**1.000**	**1.000**	**0.334**	**0.826**	**0.982**	0.677	0.820	**0.988**	0.691	**0.843**

evaluation regions. In dual context variation, default-context lags behind other approaches which shows benefit of varying context during training even in this simple setting. In the pixel modality the differences among methods are most discernible, context conditioning methods outperform domain the *hidden-context*. And among context-conditioning the *cRSSM* outperforms *concat-context* context, particularly excelling in variations of pole length and combinations of length and gravity. In the DMC Walker environment, context conditioning methods perform better than *hidden-context*. Within the featurized category, *cRSSM* takes the top spot. However, in the pixel modality, *concat-context* leads, with *cRSSM* slightly behind, except for the inter+extrapolation setting where *cRSSM* demonstrates a better understanding of the meaning of each context separately.

In summary, explicit context conditioning aids ZSG. *cRSSM* showcases improved generalization across all modalities in the Cartpole environment and delivers substantial generalization improvements in the featurized modality of DMC Walker, albeit lagging slightly behind in the pixel modality.

Finally, to conclude the section, we provide insights and visualizations supporting how our approach to context-conditioning shapes and improves Dreamer's imagination capabilities in Appendix E.

6 Conclusion and Future Work

We studied zero-shot generalization in Dreamer-style model-based reinforcement learning through the lens of contextual reinforcement learning. We discussed naive ways to incorporate contextual information into the MBRL learning pipeline and formulated the novel cRSSM for Dreamer. Our cRSSM provides a systematic approach to using context in the world modelling objectives under partial observability. Our experiments, using a rigorous evaluation protocol for zero-shot generalization, showed that naive approaches, such as domain randomization improve generalization performance. However, more principled methods such as our cRSSM are required to perform significantly better in-distribution and out-of-distribution. Our study opens the door to future work on zero-shot generalization for MBRL approaches such as Dreamer. Creating contextual benchmarks for environments such as Atari [Bellemare et al., 2013], DMLab [Beattie et al., 2016], ProcGen [Anand et al., 2021], and Minecraft [Guss et al., 2019] would be an interesting avenue for future research into ZSG, as contextual changes in these benchmarks would necessitate more pronounced changes in policies. We discuss this in more detail in Appendix I.

Our current cRSSM formulation assumes that context is observable, meaning it is directly available as input. We plan to extend the cRSSM formulation to cases where context is not directly observable

and must be inferred along with the latent states. While we show qualitative results for counterfactual dreams, the next step would be to use this to generate dreams for counterfactual contexts during training and study the effect on ZSG and sample efficiency.

References

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.

Ankesh Anand, Jacob C Walker, Yazhe Li, Eszter Vértes, Julian Schrittwieser, Sherjil Ozair, Theophane Weber, and Jessica B Hamrick. Procedural generalization by planning with self-supervised world models. In *International Conference on Learning Representations*, 2021.

M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba. Learning dexterous in-hand manipulation. *International Journal of Robotics Research*, 39(1), 2020.

P. J. Ball, C. Lu, J. Parker-Holder, and S. Roberts. Augmented world models facilitate zero-shot dynamics generalization from a single offline environment. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, volume 139 of *Proceedings of Machine Learning Research*, pages 619–629. PMLR, 2021.

A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13: 834–846, 1983.

Charlie Beattie, Joel Z. Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, Julian Schrittwieser, Keith Anderson, Sarah York, Max Cant, Adam Cain, Adrian Bolton, Stephen Gaffney, Helen King, Demis Hassabis, Shane Legg, and Stig Petersen. Deepmind lab. *ArXiv*, abs/1612.03801, 2016. URL `https://api.semanticscholar.org/CorpusID:3221395`.

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.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

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, D. Speck, S. Sievers, F. Hutter, M. Lindauer, and J. Seipp. Learning domain-independent policies for open list selection. In M. Katz, H. Palacios, and V. Gómez, editors, *Workshop on Bridging the Gap Between AI Planning and Reinforcement Learning (PRL@ICAPS'22)*, 2022.

K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems (NeurIPS'18)*, pages 4759–4770. Curran Associates, 2018.

S. Du, A. Krishnamurthy, N. Jiang, A. Agarwal, M. Dudik, and J. Langford. Provably efficient RL with rich observations via latent state decoding. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, volume 97, pages 1665–1674. Proceedings of Machine Learning Research, 2019.

Y. Duan, J. Schulman, X. Chen, P. Bartlett, I. Sutskever, and P. Abbeel. RL^2: 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.

Christian Gumbsch, Noor Sajid, Georg Martius, and Martin V. Butz. Learning hierarchical world models with adaptive temporal abstractions from discrete latent dynamics. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=TjCDNssXKU`.

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`.

William H Guss, Cayden Codel, Katja Hofmann, Brandon Houghton, Noboru Kuno, Stephanie Milani, Sharada Mohanty, Diego Perez Liebana, Ruslan Salakhutdinov, Nicholay Topin, et al. The minerl 2019 competition on sample efficient reinforcement learning using human priors. *arXiv preprint arXiv:1904.10079*, 2019.

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.

D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, volume 97, pages 2555–2565. Proceedings of Machine Learning Research, 2019.

D. Hafner, T. P. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proceedings of the International Conference on Learning Representations (ICLR'20)*, 2020. Published online: `iclr.cc`.

D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. In *Proceedings of the International Conference on Learning Representations (ICLR'21)*, 2021. Published online: `iclr.cc`.

D. Hafner, J. Pasukonis, J. Ba, and T. P. Lillicrap. Mastering diverse domains through world models. *arXiv:2301.04104 [cs.AI]*, 2023.

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

N. Hansen, H. Su, and X. Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR'24)*, 2024. Published online: `iclr.cc`.

D. Kingma and M. Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR'14)*. CBLS, 2014.

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.

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)*, 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.

C. Perez, F. P. Such, and T. Karaletsos. Generalized hidden parameter mdps: Transferable model-based rl in a handful of trials. In F. Rossi, V. Conitzer, and F. Sha, editors, *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5403–5411. Association for the Advancement of Artificial Intelligence, AAAI Press, 2020.

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.

B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Towards causal representation learning. *arXiv:2102.11107 [cs.LG]*, 2021.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *ICML*, 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.

Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. Lillicrap, and M. Riedmiller. Deepmind control suite. *arXiv preprint arXiv: 1801.00690*, 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.

P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg. Daydreamer: World models for physical robot learning. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Conference on Robot Learning (CoRL'22) 2022*, volume 205 of *Proceedings of Machine Learning Research*, pages 2226–2240. PMLR, 2022.

Chengyang Ying, Zhongkai Hao, Xinning Zhou, Hang Su, Songming Liu, Dong Yan, and Jun Zhu. Task aware dreamer for task generalization in reinforcement learning, 2024. URL `https://arxiv.org/abs/2303.05092`.

K. J. Young, A. Ramesh, L. Kirsch, and J. Schmidhuber. The benefits of model-based generalization in reinforcement learning. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, volume 202 of *Proceedings of Machine Learning Research*, pages 40254–40276. PMLR, 2023.

A. Zhang, S. Sodhani, K. Khetarpal, and J. Pineau. Learning robust state abstractions for hidden-parameter block MDPs. In *Proceedings of the International Conference on Learning Representations (ICLR'21)*, 2021a. Published online: `iclr.cc`.

B. Zhang, R. Rajan, L. Pineda, N. Lambert, A. Biedenkapp, K. Chua, F. Hutter, and R. Calandra. On the importance of Hyperparameter Optimization for model-based reinforcement learning. In A. Banerjee and K. Fukumizu, editors, *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS'21)*, pages 4015–4023. Proceedings of Machine Learning Research, 2021b.

A cRSSM Bound Derivation

The variational bound for contextual latent dynamics models $p(o_{1:H}, s_{1:H} \mid a_{1:H}, c) = \prod_t p(s_t \mid s_{t-1}, a_{t-1}, c) p(o_t \mid s_t, c)$ and a variational posterior $q(s_{1:H} \mid o_{1:H}, a_{1:H}, c) = \prod_t q(s_t \mid o_{\leq t}, a_{<t}, c)$ follows from importance weighting and Jensen's inequality as shown,

$$
\begin{aligned}
\ln p(o_{1:H} \mid a_{1:H}, c) &\triangleq \ln \mathrm{E}_{p(s_{1:H} \mid a_{1:H}, c)} \left[\prod_{t=1}^{H} p(o_t \mid s_t, c) \right] \\
&= \ln \mathrm{E}_{q(s_{1:H} \mid o_{1:H}, a_{1:H}, c)} \left[\prod_{t=1}^{H} p(o_t \mid s_t, c) p(s_t \mid s_{t-1}, a_{t-1}, c) / q(s_t \mid o_{\leq t}, a_{<t}, c) \right] \\
&\geq \mathrm{E}_{q(s_{1:H} \mid o_{1:H}, a_{1:H}, c)} \left[\sum_{t=1}^{H} \ln p(o_t \mid s_t, c) + \ln p(s_t \mid s_{t-1}, a_{t-1}, c) - \ln q(s_t \mid o_{\leq t}, a_{<t}, c) \right] \\
&= \sum_{t=1}^{H} \Big(\underbrace{\mathrm{E}_{q(s_t \mid o_{\leq t}, a_{<t}, c)} [\ln p(o_t \mid s_t, c)]}_{\text{reconstruction}} - \underbrace{\mathrm{E}_{q(s_{t-1} \mid o_{\leq t-1}, a_{<t-1}, c)} \left[\mathrm{KL}[q(s_t \mid o_{\leq t}, a_{<t}, c) \parallel p(s_t \mid s_{t-1}, a_{t-1}, c)] \right]}_{\text{complexity}} \Big).
\end{aligned}
$$

B Context Protocol

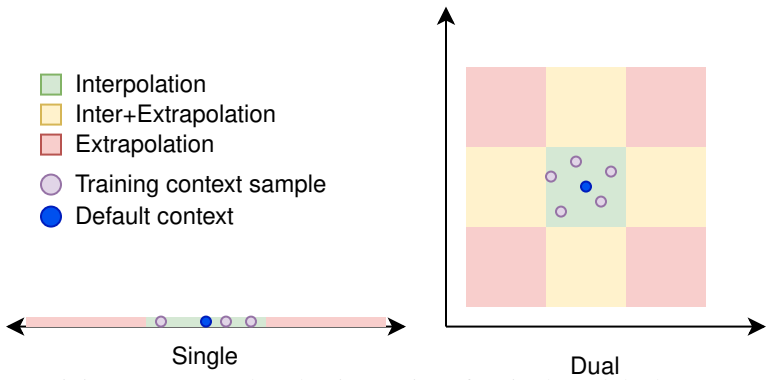


Figure 4: Training contexts and evaluation regions for single and dual context variation.

C Train and Evaluation Context Ranges

Context	Gravity	Length
Default	9.8	.5
Training Range	[4.9, 14.7]	[.35, .75]
Single Evaluation Values	.98, 17.15, 2.45, 3.92, 4.9, 7.35, 9.8, 12.25, 14.7, 15.68, 16.66, 17.64, 18.62, 19.6	.1, .15, .2, .25, .3, .4, .5, .6, .7, .8, .85, .9, .95, 1.0
Dual Evaluation Values	.98, 2.45, 3.92, 15.68, 17.64, 19.6	.1, .2, .3, .5, .7, .8, .9, 1.0

Table 2: CartPole Context Values

Context	Gravity	Actuator Strength
Default	9.8	.5
Training Range	[4.9, 14.7]	[.5, 1.5]
Single Evaluation Values	.98, 17.15, 2.45, 3.92, 4.9, 7.35, 9.8, 12.25, 14.7, 15.68, 16.66, 17.64, 18.62, 19.6	.1, .2, .3, .4, .5, .75, 1.0, 1.25, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0
Dual Evaluation Values	.98, 2.45, 3.92, 15.68, 17.64, 19.6	.1, .3, .5, 1.0, 1.5, 1.6, 1.8, 2.0

Table 3: DMC Walker Context Values

D Agent Performances

D.1 Expert and Random agent performance

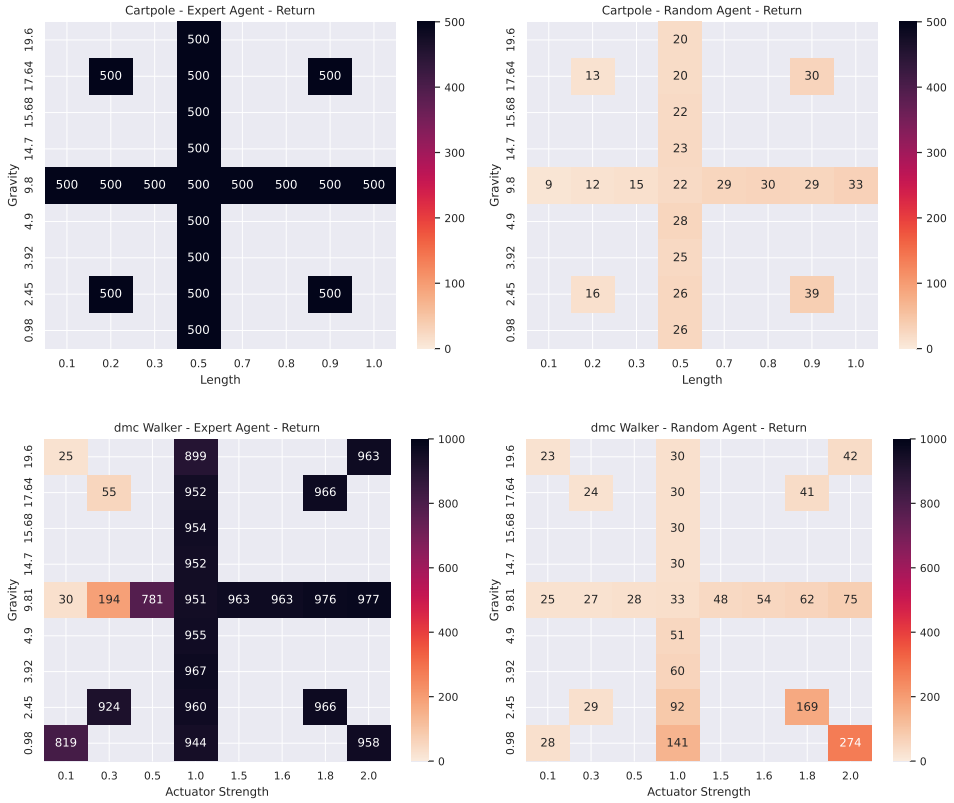


Figure 5: The best performing random policy and expert trained on each context over 5 seeds. We use featurized modality with less partial observability compared to pixels, to get an optimistic upper bound of expert returns.

D.2 Varying single context

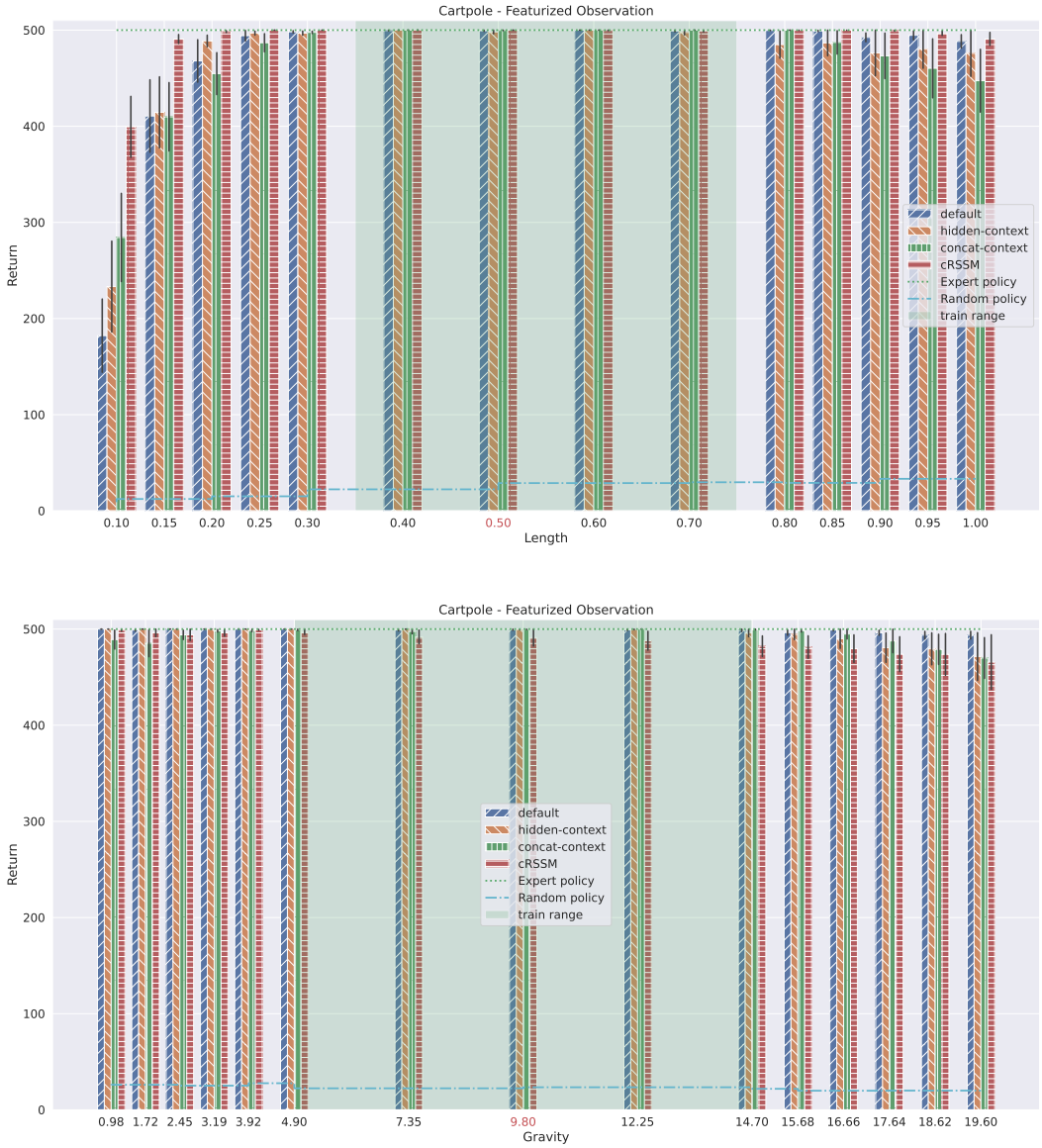


Figure 6: CartPole - Featurized Observations - The mean and standard error of the average evaluation returns are computed across 10 seeds, for 50 evaluation episodes each

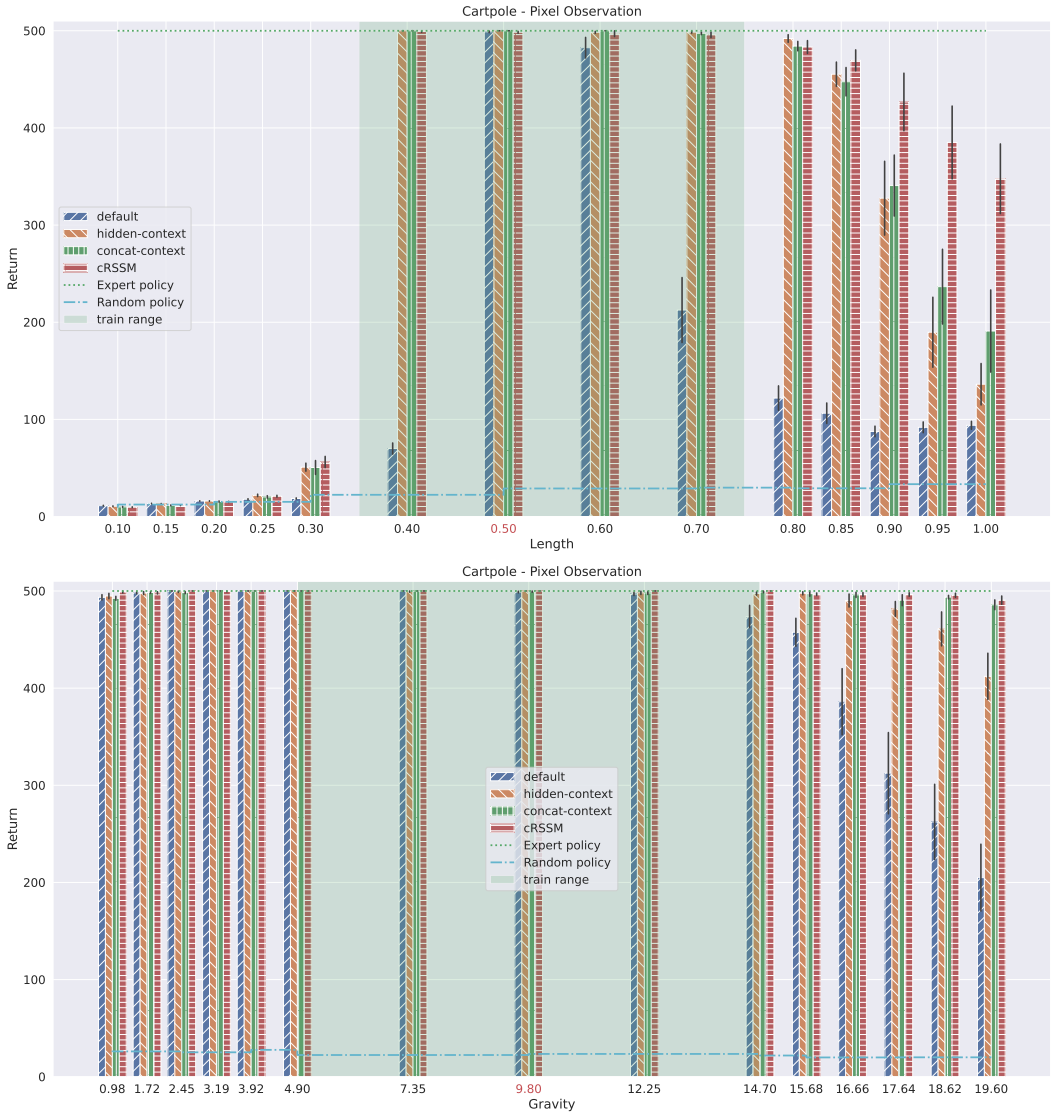


Figure 7: CartPole Pixel Observations - The mean and standard error of the average evaluation returns are computed across 10 seeds, for 50 evaluation episodes each

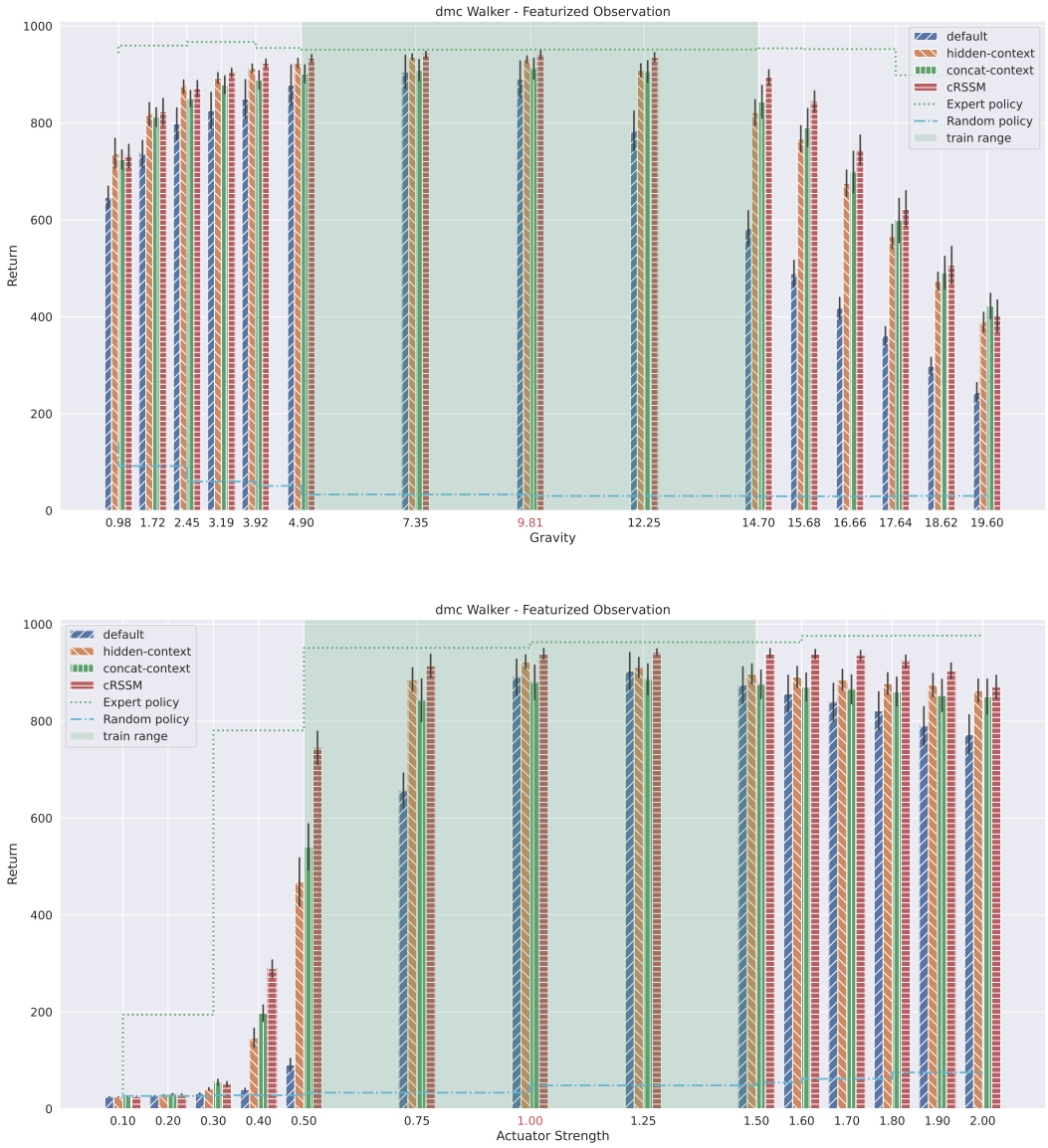


Figure 8: DMC Walker - Featurized Observations - The mean and standard error of the average evaluation returns are computed across 10 seeds, for 50 evaluation episodes each

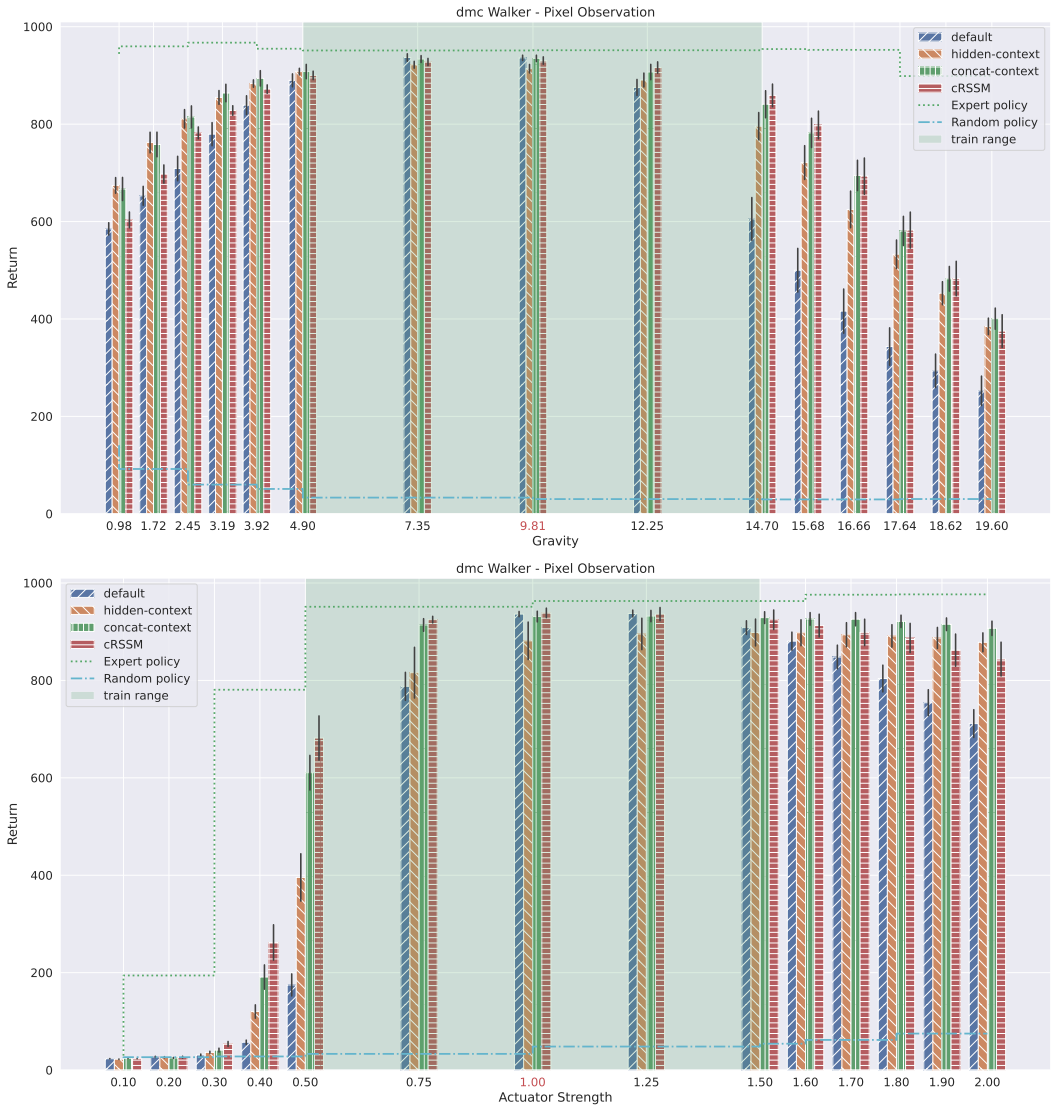


Figure 9: DMC Walker - Pixel Observations - The mean and standard error of the average evaluation returns are computed across 10 seeds, for 50 evaluation episodes each

D.3 Varying two contexts

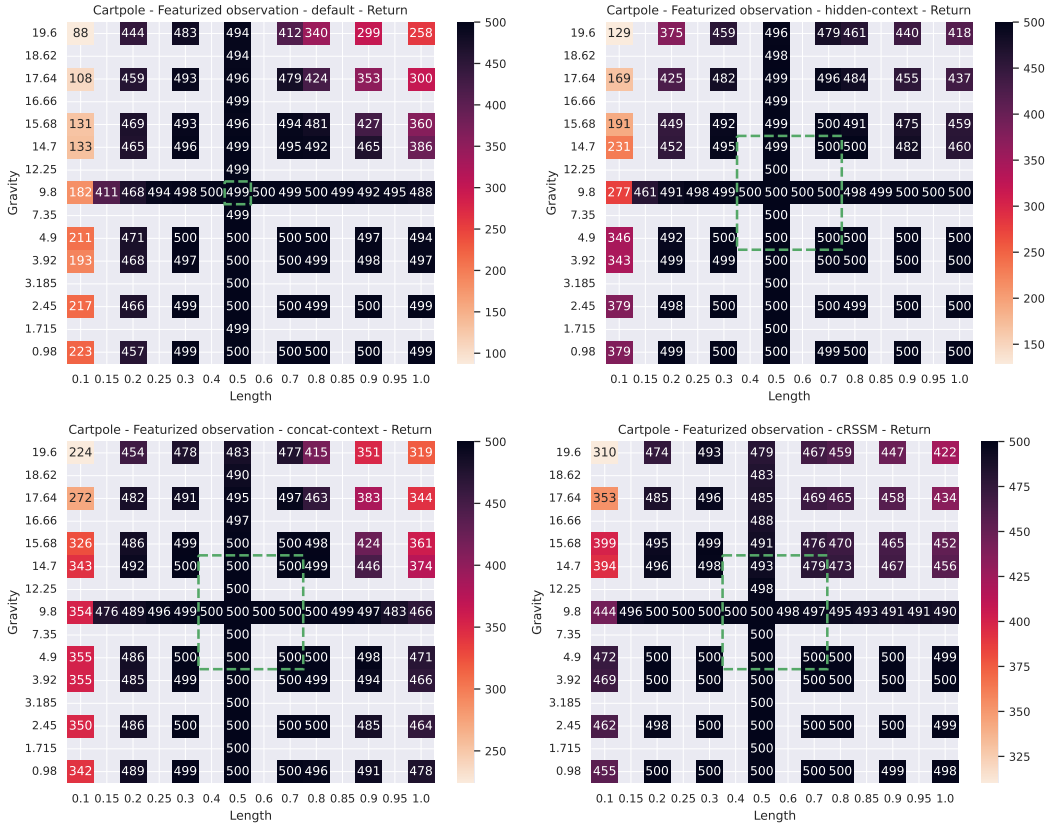


Figure 10: CartPole - Featurized - The mean and standard error of the average evaluation returns are computed across 10 seeds, for 50 evaluation episodes each

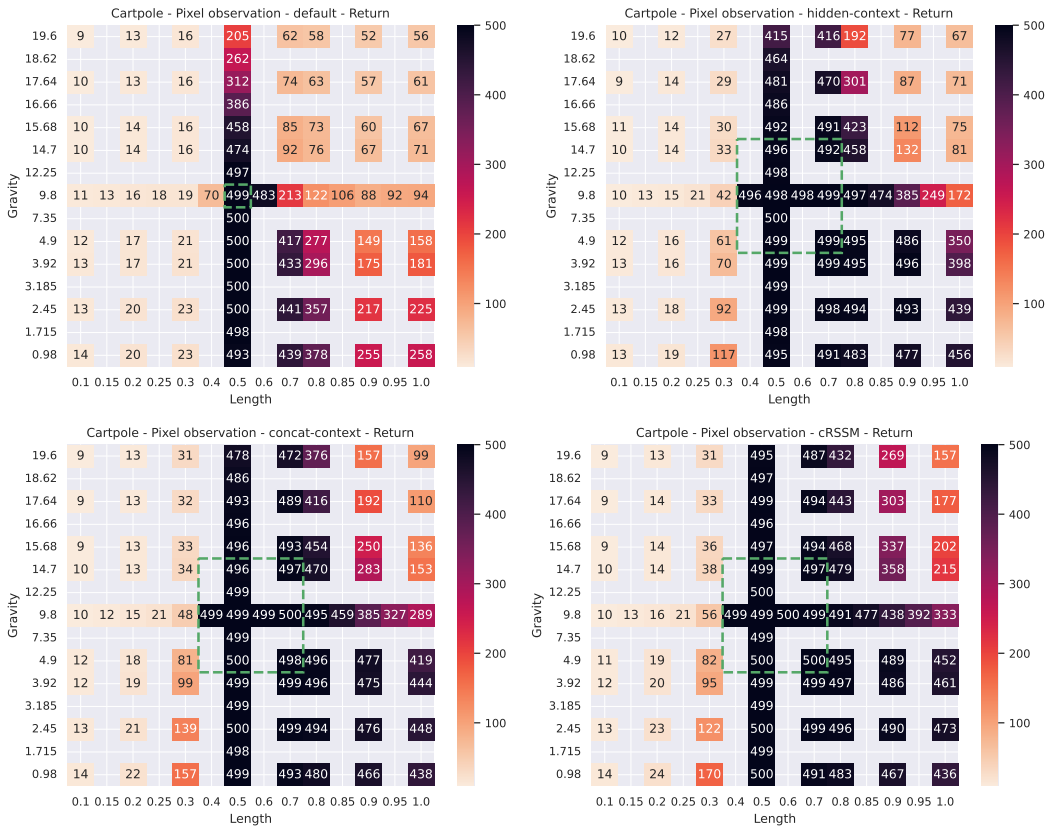


Figure 11: CartPole - Pixel - The mean and standard error of the average evaluation returns are computed across 10 seeds, for 50 evaluation episodes each

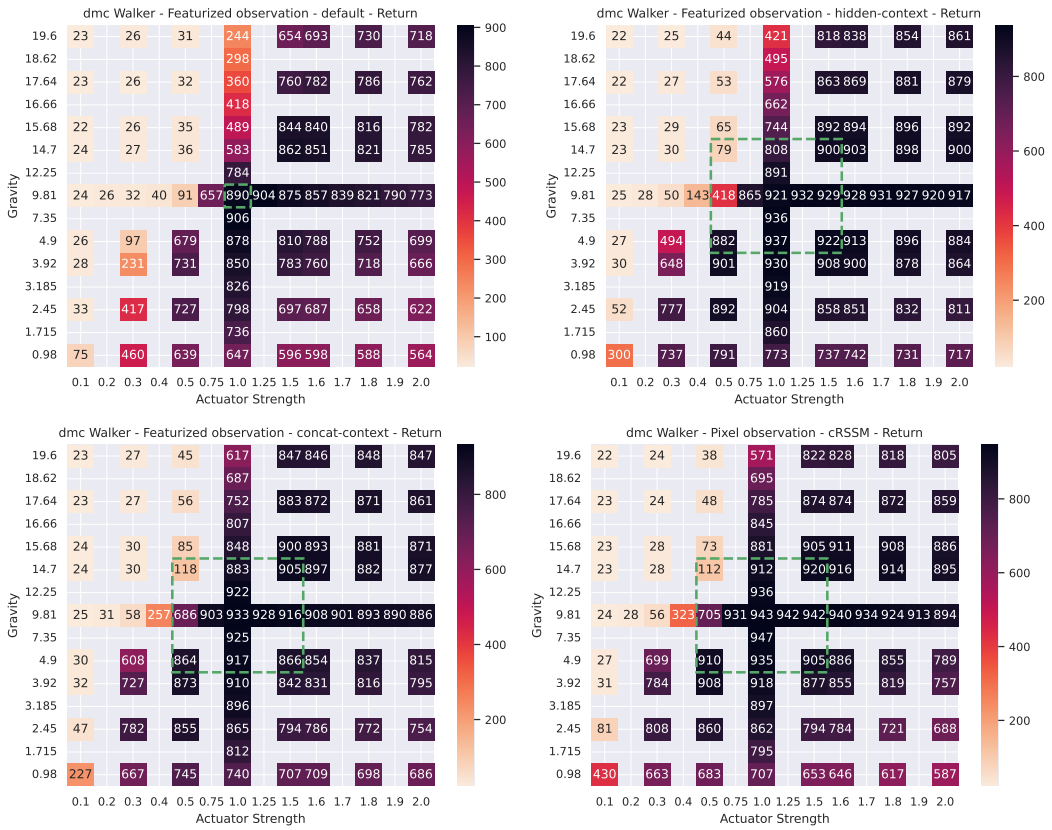


Figure 12: DMC Walker - Featurized - The mean and standard error of the average evaluation returns are computed across 10 seeds, for 50 evaluation episodes each

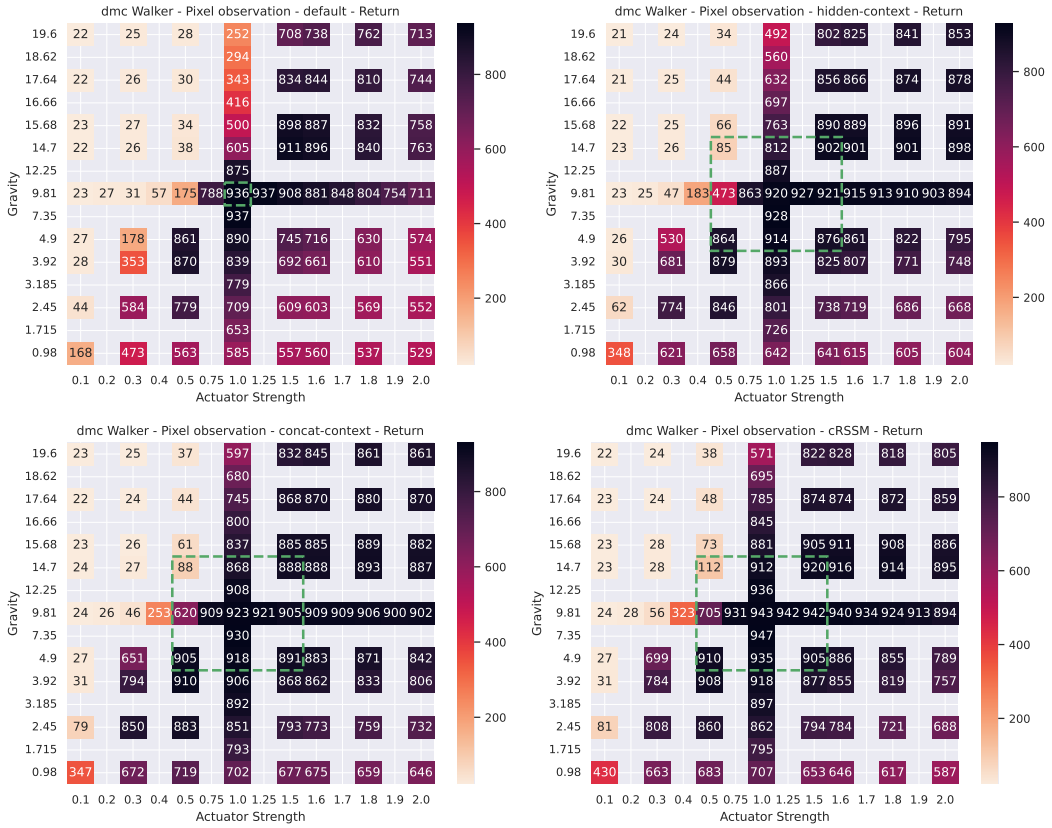


Figure 13: DMC Walker - Pixel - The mean and standard error of the average evaluation returns are computed across 10 seeds, for 50 evaluation episodes each

D.4 Probability of Improvement for cRSSM

In assessing the robustness of an algorithm's improvement over another, considering the average probability of improvement emerges as a valuable metric. Specifically, it calculates the probability of Algorithm X surpassing Algorithm Y on a randomly chosen task, disregarding the magnitude of improvement. Identifying the optimal aggregate metric remains an ongoing inquiry, and presenting multiple metrics, which circumvent the pitfalls of prevalent metrics, ensures reliability and efficiency in decision-making processes.

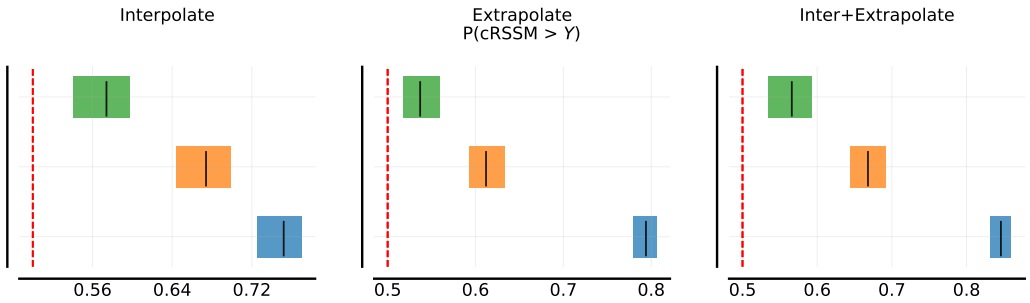


Figure 14: Aggregate probability of improvement for pixel modality.

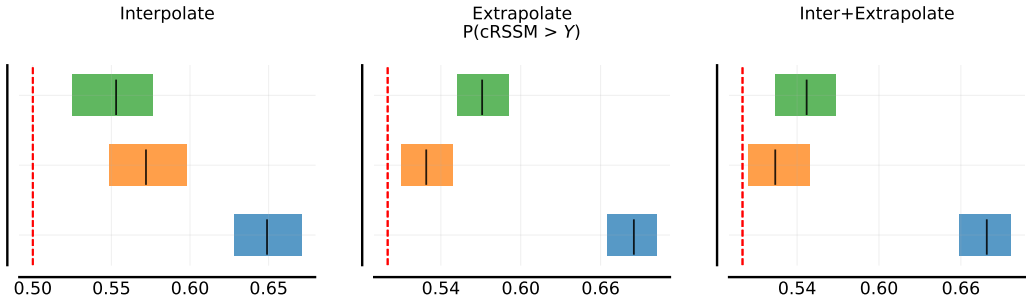


Figure 15: Aggregate probability of improvement for the featurized modality.

D.5 Expert Normalized IQM Plots for Individual Settings

The IQM plots corresponding to the settings in 1. For some settings in the Cartpole environment, since we reach optimal expert performance across all seeds, the plots look empty.

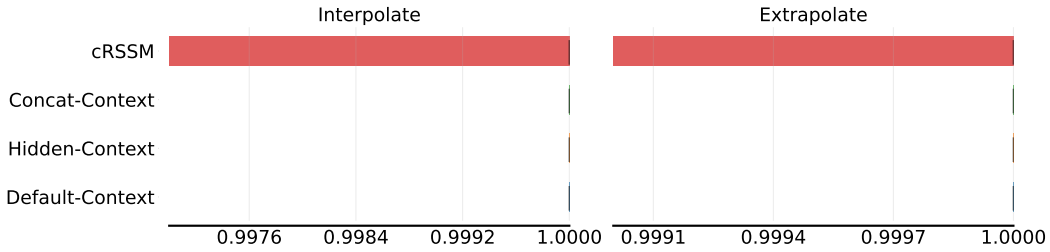


Figure 16: Cartpole - Featurized - Gravity: Expert normalized IQM with 95% confidence interval

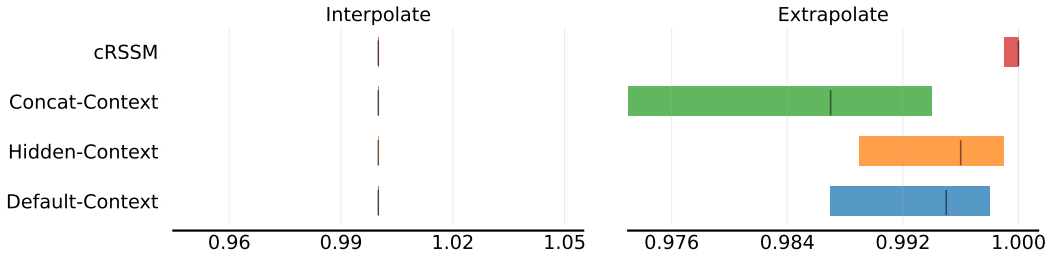


Figure 17: Cartpole - Featurized - Length: Expert normalized IQM with 95% confidence interval

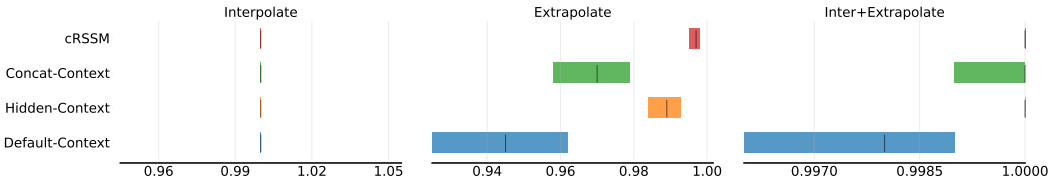


Figure 18: Cartpole - Featurized - Gravity + Length: Expert normalized IQM with 95% confidence interval

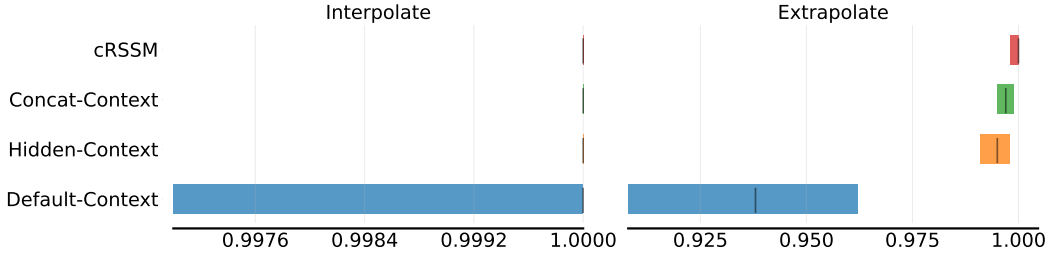


Figure 19: Cartpole - Pixel - Gravity: Expert normalized IQM with 95% confidence interval

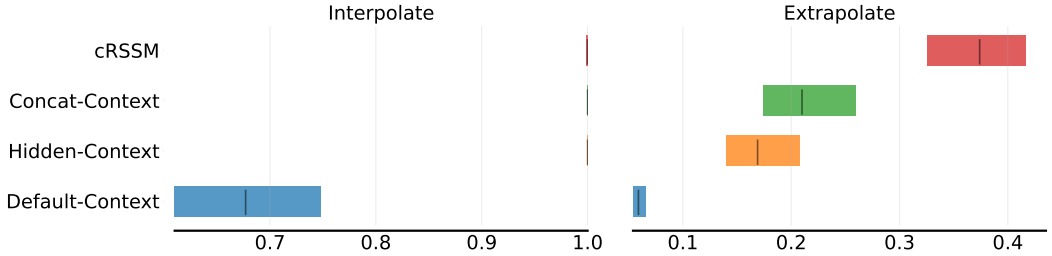


Figure 20: Cartpole - Pixel - Length: Expert normalized IQM with 95% confidence interval

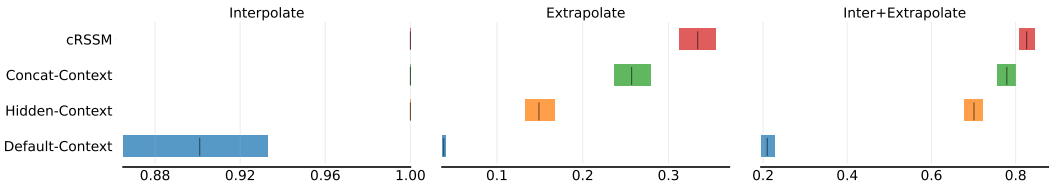


Figure 21: Cartpole - Pixel - Gravity + Length: Expert normalized IQM with 95% confidence interval

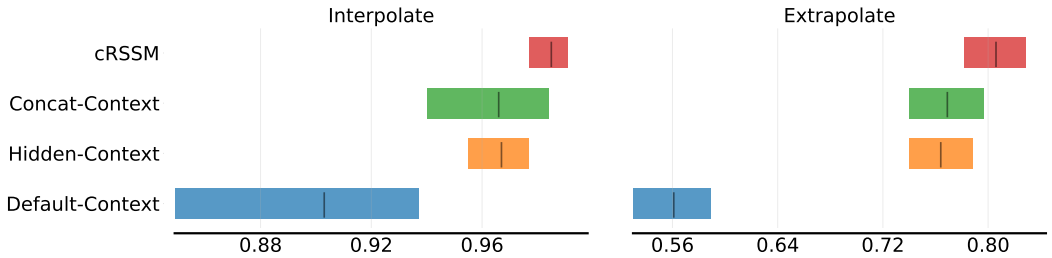


Figure 22: DMC Walker - Featurized - Gravity: Expert normalized IQM with 95% confidence interval

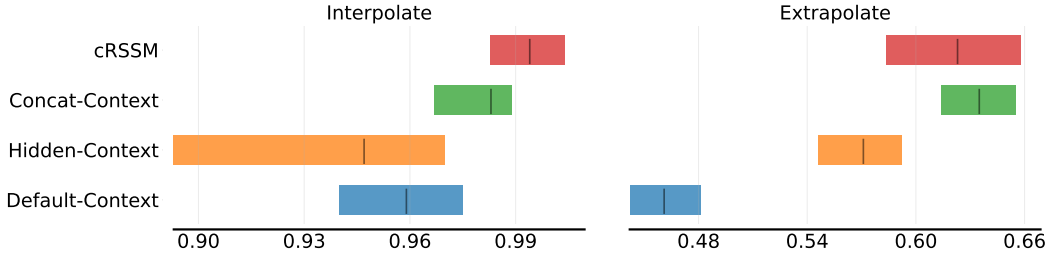


Figure 23: DMC Walker - Featurized - Actuator Strength: Expert normalized IQM with 95% confidence interval

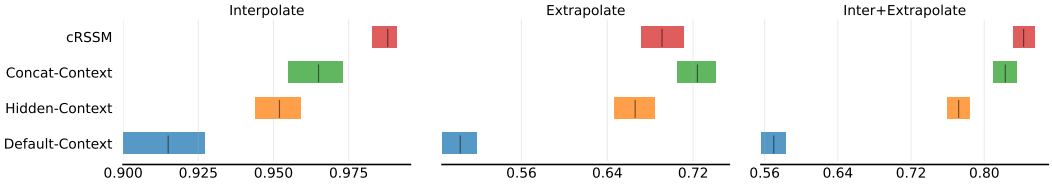


Figure 24: DMC Walker - Featurized - Gravity + Actuator Strength: Expert normalized IQM with 95% confidence interval

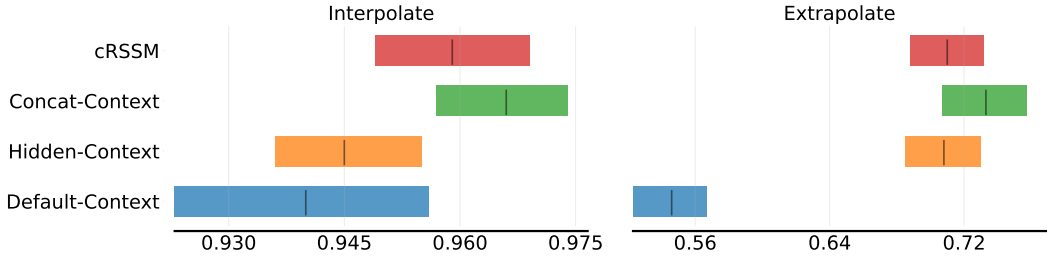


Figure 25: DMC Walker - Pixel - Gravity: Expert normalized IQM with 95% confidence interval

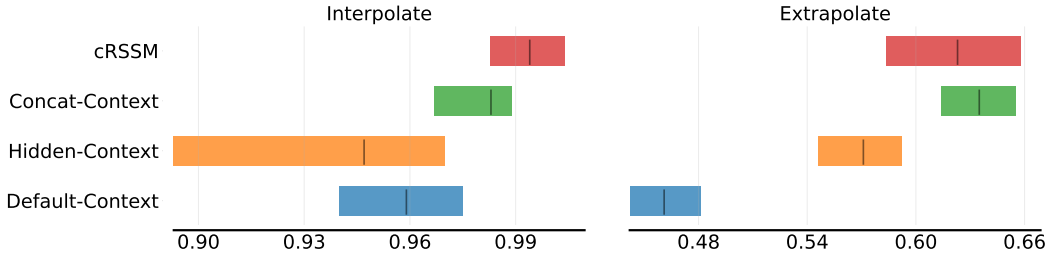


Figure 26: DMC Walker - Pixel - Actuator Strength: Expert normalized IQM with 95% confidence interval

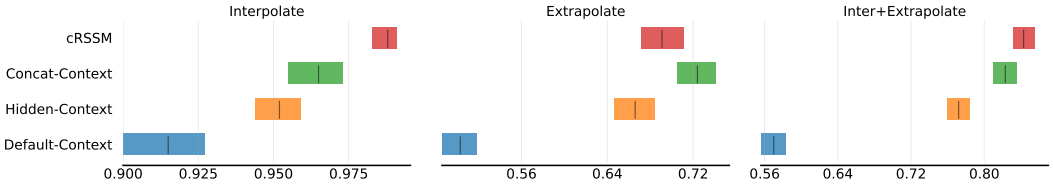


Figure 27: DMC Walker - Pixel - Gravity + Actuator Strength: Expert normalized IQM with 95% confidence interval

E Beyond Task Performance, Does Context-Conditioning Impact the Latent States?

We qualitatively assess the ability of different methods to understand context and its impact on latent representations to help explain differences in performance and shortcomings of different approaches.

To evaluate how the methods use context, we visually investigate Dreamer's reconstruction of visually observable OOD contexts, which are either encoded into the latent state in concat-context or conditioned separately cRSSM. We choose the visually observable context to be the pole length in Cartpole, evaluating lengths shorter and longer than what it has seen in the training distribution (labeled short and long in Figure 28). A model capable of generating images conditioned on novel context demonstrates a semantic understanding of the context that is well grounded in the image space. In cases involving context-conditioned models, we also provide counterfactual visual explanations, exploring how the reconstructed pixel observation $\hat{o}_t$ inferred from the original observation that encodes the factual context (c_F) differs if conditioned on a counterfactual context (c_{CF}).

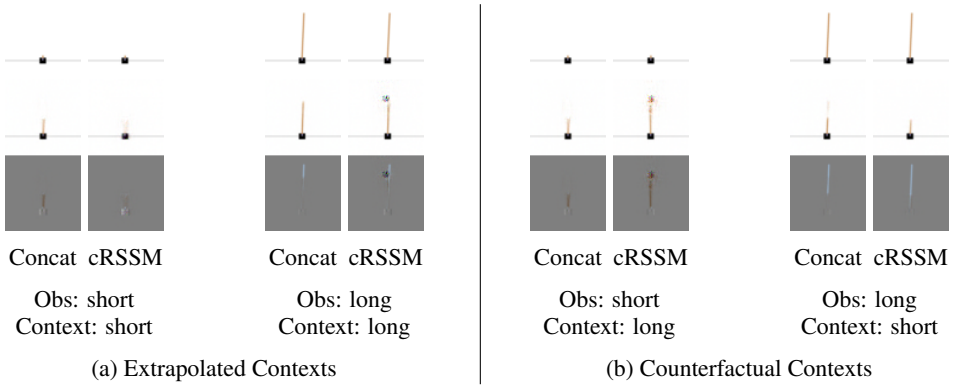


Figure 28: Qualitative results for the model generative ability of novel context. In each image, we have the true observation, followed by the one reconstructed by the decoder with context conditioning from the latent encoded from the true image, and lastly the difference between the two images. The *short* refers to a length of 0.1 units, and *long* is the OOD length of 1.0 units. In the extrapolation case, ideally, the difference should be minimal and, in the counterfactual case maximal.

Extrapolation In the extrapolation case depicted in Figure 28a, we encode observations from the OOD pole length context (short: 0.1 or long: 1.0) and also condition on the true OOD context value. It can be seen that the cRSSM predicted observation is more faithful to the OOD contexts compared to other methods. In the case of short length, cRSSM generates a slightly blurred and shorter pole, while concat-context is confined to the shortest in-distribution pole. For the longer pole, cRSSM exhibits more realistic behavior by attempting to add additional pixels on top of the longest pole it has seen, demonstrating a better semantic understanding of the context by the world model. In contrast, concat-context falls short and instead decodes the longest in-distribution pole length. We attribute the bottleneck that confines concat-context to the bounds of the context seen during training to the discretization of the latent states (containing the context). Notably, the decoder's ability to decode shorter-length poles for shorter inputs and vice versa suggests that the encoder of both methods has learned the scale of pole lengths meaningfully.

Why Can Assessing Disentanglement Help? Although we see a meaningful extrapolation, the length and pole position could also have been encoded into the latent state. This would defeat the purpose of conditioning on the context. Ideally, we want the system to interpret our context as the source of truth and not redundantly encode it from observations. The "sparse mechanism shift" hypothesis [Schölkopf et al., 2021] states that such disentanglement of causal mechanisms in representations enables OOD generalization.

Counterfactual Assessment of Disentanglement To test for disentanglement and faithfulness to the conditioning context, we use the ability of our cRSSM world model to dream of many worlds by taking observations generated from the factual context c_F and encoding them to the latent state while conditioning the model in the counterfactual context c_{CF}. Then, we decode the image to see how counterfactual conditioning influences image generation. From Figure 28b, we can see that cRSSM uses the context value more faithfully than concat-context. This demonstrates the capability of cRSSM to extrapolate and combine the conditioning context with the latent state to generate semantically meaningful counterfactual images. In contrast to the context-disentangled latent space of cRSSM, the concat-context approach encodes both context information and observations jointly into the latent state, hindering its ability to generalize effectively.

Our investigation reveals clear evidence of extrapolation capabilities in our proposed principled cRSSM approach compared to the vanilla concat-context strategy. Furthermore, through our visual counterfactual explanations, we observe indications that the latent state in cRSSM appears disentangled from the context, which explains the observed gains in generalization using this approach.

F Intuitive Interpretation of the RSSM & cRSSM

We can use a video game analogy to gain an intuitive understanding of the RSSM and cRSSM. The context is the game settings, such as difficulty, which do not change while playing the game. The deterministic state is the memory of the game engine. The stochastic state models the aleatoric uncertainty used by the game engine, i.e. whenever the game samples from a random number generator and certain variables of the current game state, it is the same as sampling from the stochastic state model. After this sampling is done and the user inputs the action, the deterministic state model is akin to the game's logic, which uses the context of the current game memory state and the sampled stochastic state to compute the next game memory state. The observation model is the game engine's visual renderer that maps the game memory to the pixels you see on your monitor. It can also use context to render things differently. Finally, the reward model is the score, a distribution conditioned on the state and context, as the context (say, difficulty setting) can influence how many points you get for a given state.

G Hyperparameters

We choose the *small* variant of DreamerV3 with all hyperparameters taken from Hafner et al. [2023].

Name	**Value**
General	
Replay capacity (FIFO)	10^6
Batch size	16
Batch length	64
Activation	$\mathrm{LayerNorm} + \mathrm{SiLU}$
World Model	
Deterministic State model (GRU) units	512
MLP layers	2
MLP units	512
Number of latents	32
Classes per latent	32
Reconstruction loss scale	1.0
Dynamics loss scale	0.5
Representation loss scale	0.1
Learning rate	10^{-4}
Adam epsilon	10^{-8}
Gradient clipping	1000
Actor Critic	
MLP layers	2
MLP units	512
Imagination horizon	15
Discount horizon	333
Return lambda	0.95
Critic EMA decay	0.98
Critic EMA regularizer	1
Return normalization scale	$\mathrm{Per}(R, 95) - \mathrm{Per}(R, 5)$
Return normalization limit	1
Return normalization decay	0.99
Actor entropy scale	$3 \cdot 10^{-4}$
Learning rate	$3 \cdot 10^{-5}$
Adam epsilon	10^{-5}
Gradient clipping	100

Table 4: DreamerV3 hyper parameters. The same values are used across all experiments.

H Results for DMC Walker - 100k Steps

For comprehensive evaluations, we conducted intermediate assessments on the DMC walker environment, using 10 seeds for 100k environment steps in each generalization setting. Normalized IQM scores, detailed in Table 5, demonstrate superior performance in the most challenging featurized cases with both contexts. However, within some settings, particularly for the pixel modality, we observed a notable lag in performance. This discrepancy, especially in the interpolation region, where the evaluation distribution aligns closely with the training distribution, indicates the need for additional samples to facilitate effective learning. We present the complete training (500k steps) in Table 1.

	Walker - 100k steps					
	Featurized			Pixel		
(g d)	0.737	0.551	-	0.549	0.376	-
(g h)	0.682	0.547	-	**0.697**	0.489	-
(g c)	**0.864**	**0.684**	-	0.656	**0.520**	-
(g cR)	0.779	0.661	-	0.565	0.450	-
(a d)	0.824	0.406	-	0.634	0.305	-
(a h)	0.833	0.437	-	0.674	0.376	-
(a c)	0.741	0.402	-	0.732	**0.409**	-
(a cR)	**0.947**	**0.456**	-	**0.819**	0.391	-
(g+a d)	0.749	0.411	0.537	0.574	0.326	0.372
(g+a h)	0.722	0.401	0.521	0.649	0.396	0.469
(g+a c)	0.652	0.357	0.470	**0.658**	**0.406**	**0.494**
(g+a cR)	**0.884**	**0.447**	**0.632**	0.606	0.333	0.423

Table 5: Expert normalized IQM over 10 seeds for different evaluation settings, in featurized and pixel modality. Each described by three variables: context, method, and mode. Context takes values from $\{g : \text{gravity}, a : \text{actuator strength}, l : \text{pole length}\}$ with + indicating multiple contexts; and method from $\{d : \text{default-context}, h : \text{hidden-context}, c : \text{concat-context}, cR : \text{cRSSM}\}$

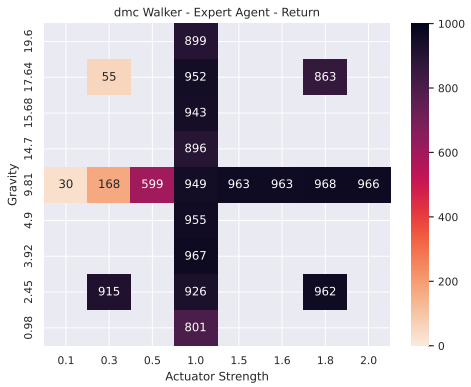


Figure 29: The best expert trained on each context over 5 seeds for 100k steps on DMC walker. We use featurized modality with less partial observability compared to pixels, to get an optimistic upper bound of expert returns.

I Discussion on More Comprehensive Benchmarks

Creating contextual benchmarks for environments such as Atari, DMLab, ProcGen, and Minecraft presents an important opportunity for further research into ZSG. Unlike our current tasks focusing on motor control in environments like Walker and CartPole, some of these benchmarks emphasize different aspects such as navigation and exploration while others such as procedurally generated or open-ended worlds offer dynamic objectives and high variability, requiring strategic planning and

adaptability to diverse challenges. These complexities necessitate pronounced policy adjustments with changing contexts. However, creating variants in such benchmarks is challenging due to their inherent intricacies and lack of easily accessible interfaces. By addressing these complexities, future work could unlock critical insights into the adaptability of reinforcement learning agents in diverse and changing conditions.