
Dynamics-Aligned Shared Hypernetworks for Contextual RL under Discontinuous Shifts

Jan Benad^{1*} Pradeep Kr. Banerjee^{1*} Frank Röder¹
Nihat Ay^{1,2} Martin V. Butz³ Manfred Eppe¹

¹Institute for Data Science Foundations, TUHH, Germany

²Santa Fe Institute, USA

³Neuro-Cognitive Modeling Group, University of Tübingen, Germany

Abstract

Zero-shot generalization in contextual reinforcement learning remains a core challenge, particularly when the context is latent and must be inferred from data. A canonical failure mode arises when latent context discontinuously changes how actions affect the environment, requiring incompatible control responses across contexts. We propose DMA*-SH, a framework where a single hypernetwork, trained solely via dynamics prediction, generates a small set of adapter weights shared across the dynamics model, policy, and action-value function. This shared modulation imparts an inductive bias matched to discontinuous context-to-dynamics shifts, while input/output normalization and random input masking stabilize context inference, promoting directionally concentrated representations. We provide theoretical support via expressivity separation results for hypernetwork modulation, and a variance decomposition with policy-gradient variance bounds that formalize how within-mode compression improves learning under non-overlapping contexts. For evaluation, we introduce the Actuator Inversion Benchmark (AIB), a suite of environments designed to isolate challenging context-to-dynamics interactions, including actuator inversion, actuator permutations, and weakly non-overlapping continuous dynamics. On AIB’s held-out tasks, DMA*-SH achieves zero-shot generalization, outperforming domain randomization by 58.1% and surpassing a standard context-aware baseline by 11.5% on average.

1 Introduction

Imagine switching to a colleague’s laptop and discovering the trackpad scrolls in the opposite direction. The screen moves *against* your fingers instead of with them. Despite years of motor expertise, you are suddenly disoriented. Every instinctual swipe produces the wrong outcome. Trying harder does not help; you must override muscle memory. This scenario captures a core challenge in contextual reinforcement learning (RL): zero-shot adaptation can fail when context induces discontinuous changes in action effects.

The trackpad scenario is an instance of what we call *actuator inversion*, and serves as a minimal example of a broader problem: latent context can change the action-effect map discontinuously. The same difficulty arises in swapped control channels, coordinate remappings, gain changes, or sim-to-real actuator mismatches. In such settings, there may be no valid interpolation between incompatible control laws; an “averaged” behavior can be incoherent. Smooth inductive biases, such as context concatenation or Gaussian structure in latent context, can then blur distinct modes into an averaged representation. Standard domain randomization (DR) can similarly yield policies that compromise across regimes and degrade performance in both. These failures reflect structural mismatches between

*Equal contribution

architectural inductive biases and the problem’s inherent discontinuity. Zero-shot adaptation to such context shifts requires representing and executing multiplicative switching reliably. In practice, this is difficult when context is inferred from short, noisy windows, since small inference errors can trigger the wrong mode and yield unstable behavior.

Such discontinuous shifts require a specific inductive primitive: *multiplicative* context-dependent modulation of network parameters, which can represent functional opposites, for example by inducing approximately sign-reversed or permuted adaptations across modes. Hypernetworks [Ha et al., 2017, Jayakumar et al., 2020] provide a natural mechanism for this modulation. Building on this principle, we propose DMA*-SH (Dynamics Model-Aligned, Shared Hypernetwork), a framework for context-inferred RL where a single hypernetwork, trained solely via forward dynamics prediction, generates a small set of adapter weights shared across the dynamics model, policy, and Q-function. This design couples control to dynamics-consistent modulation and supports discontinuous, multiplicative regime switches. Carefully chosen normalizations and input masking further encourage directional concentration of context information and are associated with stable representations that compress irrelevant continuous variations while preserving task-critical discontinuous structure.

This paper makes the following contributions:

- **Actuator Inversion Benchmark (AIB).** We introduce AIB, a suite of contextual RL environments designed to isolate challenging context-to-dynamics interactions, including actuator inversion, actuator permutations, and weakly non-overlapping continuous dynamics, while still encompassing standard physical parameter variations.
- **DMA*-SH Architecture.** We propose a shared-hypernetwork framework for context-inferred RL that provides multiplicative modulation for discontinuous context-to-dynamics shifts and couples adaptation to dynamics-consistent conditioning through shared adapters across modules.
- **Theoretical Analysis.** We provide (i) an expressivity separation result for hypernetwork modulation (Theorems A.1, A.2 and A.5) formalizing the multiplicative advantage of DMA*-SH for discontinuous contexts; (ii) a variance decomposition (Theorem A.15) that isolates distinct sources of embedding variance under actuator inversion; and (iii) a policy-gradient variance bound (Theorem A.17) linking within-mode compression to learning stability, motivating a structural information bottleneck view.
- **Empirical validation.** On AIB’s held-out, DMA*-SH achieves zero-shot generalization, outperforming domain randomization by 58.1%, and context-aware baselines Concat (concatenation-based and known context) by 11.5%, and DA (with separate policy/value hypernetworks and known context) by 4.6% on average. Ablations confirm that normalization, masking, and hypernetwork sharing each contribute to performance. Code is available at: <https://github.com/dma-sh/dmash>.

2 Background

Contextual reinforcement learning. We formalize the problem using the *Contextual Markov Decision Process* (CMDP) framework [Hallak et al., 2015, Benjamins et al., 2023]. A CMDP is defined by the tuple $(\mathcal{C}, \mathcal{S}, \mathcal{A}, \{P^c\}, \{r^c\}, \gamma)$, where \mathcal{C} is the context space, \mathcal{S} and \mathcal{A} are the state and action spaces, $P^c(s'|s, a)$ specifies the transition probability from state s to s' under action a in context $c \in \mathcal{C}$, $r^c: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. Each context c defines a distinct MDP with shared \mathcal{S} and \mathcal{A} but possibly differing dynamics P^c and reward r^c . The context is fixed within an episode. Following prior work [Beukman et al., 2023, Benjamins et al., 2023, Prasanna et al., 2024, Röder et al., 2025], we focus on variations in transition dynamics only, keeping the reward fixed: $r^c = r$ for all $c \in \mathcal{C}$.

Zero-shot generalization. To evaluate generalization, we define three context sets [Kirk et al., 2023]: $\mathcal{C}_{\text{train}}$ for training, $\mathcal{C}_{\text{eval-in}}$ for in-distribution (interpolation) evaluation, and $\mathcal{C}_{\text{eval-out}}$ for out-of-distribution (extrapolation) evaluation. Each set consists of context instances sampled from environment-specific ranges; sampled instances are pairwise disjoint across sets. The agent learns a policy π_θ maximizing expected return over training contexts: $\frac{1}{|\mathcal{C}_{\text{train}}|} \sum_{c \in \mathcal{C}_{\text{train}}} \mathbb{E}_{\pi_\theta} [\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$, where $s_{t+1} \sim P^c(\cdot | s_t, a_t)$. During zero-shot evaluation, the agent receives no gradient updates.

3 Related Work

Contextual RL provides a natural framework for studying zero-shot generalization across varying environments [Hallak et al., 2015, Modi et al., 2018, Beck et al., 2023a, Kirk et al., 2023]. Prior work broadly assumes either observable context variables [Chen et al., 2018, Seyed Ghasemipour et al., 2019, Ball et al., 2021, Eghbal-zadeh et al., 2021, Sodhani et al., 2021, Mu et al., 2022, Benjamins et al., 2023, Prasanna et al., 2024] or infers context implicitly from interaction history [Chen et al., 2018, Xu et al., 2019, Lee et al., 2020, Seo et al., 2020, Xian et al., 2021, Sodhani et al., 2022, Melo, 2022, Evans et al., 2022, Ndir et al., 2024, Röder et al., 2025]. We follow the latter and learn context representations via self-supervised alignment with a dynamics model, without assuming explicit context access. Although recurrent agents can acquire latent contextual representations [Grigsby et al., 2024a,b, Luo et al., 2024, Hafner et al., 2019, 2025], these representations are typically not grounded in environment dynamics. Closely related, Beukman et al. [2023] employ hypernetworks [Ha et al., 2017] conditioned on explicitly provided context variables rather than inferring context from experience; in contrast, we train a single dynamics-aligned hypernetwork shared across policy and value functions, enabling unified context inference for zero-shot generalization. An extended discussion of contextual RL and related approaches is provided in Appendix B.

4 Context Encoding and Utilization

We present a framework for learning a **dynamic model-aligned (DMA)** context representation. We first introduce **DMA***, which incorporates key enhancements to vanilla DMA that refine and stabilize this latent representation. We then present our shared-hypernetwork approach for incorporating latent context information. We refer to this method as **DMA*-SH**, as it extends **DMA*** with a shared hypernetwork that jointly informs the dynamics model, policy, and action-value function.

4.1 Context Inference by Dynamic Model-Aligned Representation Learning

We denote by τ_t^c a sliding window of the past K transitions from the same context c , each given as a tuple $(s_t, a_t, \delta s_{t+1})$, where $\delta s_{t+1} = s_{t+1} - s_t$ is the state difference. The sequence τ_t^c is passed through a *context encoder* $g_\phi(\tau_t^c)$ to produce a context representation $z_t \in \mathbb{R}^{d_z}$. The context encoder is trained jointly with a forward dynamics (FD) model f_θ that predicts the next state difference $\delta \hat{s}_{t+1}$ given the current state s_t , action a_t , and inferred context z_t . The objective is a reconstruction loss between predicted and true next state differences:

$$L_{\phi, \theta} = \|\delta \hat{s}_{t+1} - \delta s_{t+1}\|_2^2. \tag{1}$$

This defines the vanilla **DMA** context-encoding objective. Next, we describe two key modifications that improve the quality and robustness of the learned context representations: random input masking and input/output normalization. We refer to this enhanced context encoder as **DMA***; see Figure 9 (Appendix D) for the full encoder pipeline. Ablations for our design choices appear in Appendix F.

Input masking. Random masking of input features has been shown to improve representation learning across vision, language, and decision-making domains [Devlin et al., 2019, Liu et al., 2022, He et al., 2022]. We use masking purely as input corruption, rather than optimizing a masked-prediction objective. During training, we apply random masking independently at each timestep within the K -step window: for each tuple $(s_t, a_t, \delta s_{t+1})$, we independently zero out the state, action, and next-state-difference vectors with a fixed masking probability. Masking reduces reliance on brittle feature co-adaptations and discourages spurious correlations that do not generalize across contexts.

Input normalization. After masking, the concatenations of $(s_t, a_t, \delta s_{t+1})$ from τ_t^c are processed by a linear layer and normalized via AvgL1Norm [Fujimoto et al., 2023], defined as $\text{AvgL1Norm}(x) = \frac{Nx}{\sum_i |x_i|}$. Unlike BatchNorm [Ioffe and Szegedy, 2015], which relies on running statistics and can degrade under small-batch online RL, AvgL1Norm is a per-sample, statistic-free operator suited to sliding windows. It prevents monotonic growth in representation space while preserving relative feature scales [Gelada et al., 2019], yielding consistent embeddings across $\mathcal{C}_{\text{train}}$ and $\mathcal{C}_{\text{eval-out}}$.

Output normalization. The normalized and masked sequence is processed by an LSTM, and the final hidden state is projected to a compact context embedding $z_t \in \mathbb{R}^{d_z}$ with $d_z = 8$. We normalize z_t

using SimNorm [Lavoie et al., 2023, Hansen et al., 2024], which projects z_t into $L = 2$ distinct $V = 4$ dimensional simplices via a group-wise softmax. By constraining the representation to a product of simplices, SimNorm stabilizes online RL by bounding the representation scale and promoting sparsity without relying on batch statistics, preventing representation collapse and improving sample efficiency [Obando-Ceron et al., 2025].

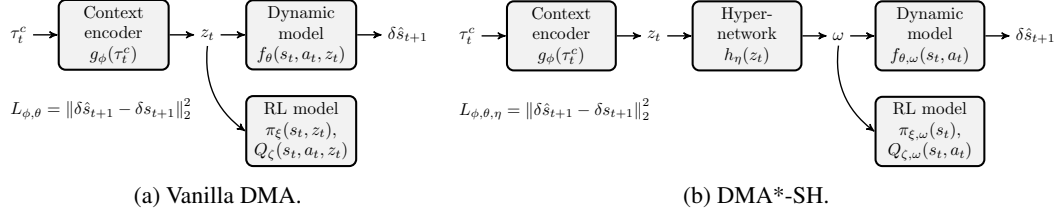


Figure 1: (a) In vanilla **DMA**, the inferred context z_t is concatenated to the RL inputs. (b) In **DMA*-SH**, a hypernetwork h_η conditioned on z_t generates adapter weights ω that are used by the forward dynamics model and the RL networks. The context encoder and hypernetwork are trained via the reconstruction objective $L_{\phi,\theta,\eta}$ in (2), while during RL updates gradients through $\omega = h_\eta(z_t)$ are stopped so that the policy and critic losses do not backpropagate to η (or to z_t) through the shared adapter pathway $z_t \rightarrow \omega \rightarrow (\pi, Q)$.

4.2 Context Utilization by a Shared Dynamic Model-Aligned Hypernetwork (DMA*-SH)

In the vanilla DMA setup, the policy and Q-function receive the concatenation of the state s_t and the inferred context z_t as input (Figure 1a). In contrast, we incorporate z_t using a hypernetwork [Ha et al., 2017], which is a meta- or second-order neural network [Pollack, 1990, Sugita et al., 2011, Beukman et al., 2023] that generates weights for a target network in an end-to-end differentiable manner. In our approach, the hypernetwork generates weights for only a subset of the main network. We refer to these second order parametrized parts as *adapters*.

As described in Section 4.1, the context representation z_t is first inferred from past transitions in τ_t^c via dynamic model-aligned representation learning. A hypernetwork h_η is then conditioned on z_t to produce weights ω for the adapters in the dynamic model $f_{\theta,\omega}$, whose parameters are therefore split into generated weights ω and remaining base weights θ . The parameters ϕ , θ , and η for the context encoder, dynamics model, and hypernetwork are updated jointly using the reconstruction loss:

$$L_{\phi,\theta,\eta} = \|\delta\hat{s}_{t+1} - \delta s_{t+1}\|_2^2. \quad (2)$$

Finally, without further modification, the generated adapter weights $\omega = h_\eta(z_t)$ are reused by the policy and Q-function (with base parameters ξ and ζ and adapter weights ω) via the adapter pathway $z_t \rightarrow \omega \rightarrow \{\pi_{\xi,\omega}, Q_{\zeta,\omega}\}$ (Figure 1b). During RL updates, we detach ω in the actor and critic losses to prevent reward gradients from modifying the shared hypernetwork and the adapter mapping $z_t \mapsto \omega$; consequently, these gradients also do not reshape the context embedding z_t through the adapter pathway. Thus, this mapping (and the induced context representation it relies on) is trained exclusively through the reconstruction loss (2), embedding dynamics-aligned features into ω . This constraint acts as a structural prior, requiring the actor and critic to process context through adapters shaped by the dynamics. This dynamics-grounded factorization separates mode identification from mode-conditioned control, unlike recurrent or Transformer agents that entangle both in an RL-trained hidden state (Appendices A.5 and A.6). For detailed pseudocode, see Algorithms 1 and 2 in Appendix C.

4.3 Expressive Advantage of DMA*-SH via Multiplicative Adapters

A key architectural advantage of DMA*-SH over vanilla DMA arises from using hypernetwork-conditioned *multiplicative adapters* rather than simple concatenation. In vanilla DMA, context enters the actor and critic through the input channel, via concatenated vectors such as $[s_t; z_t]$ and $[s_t; a_t; z_t]$. In this case, any context-dependent transformation of internal features must be synthesized implicitly by the downstream network from additive access to z_t . In contrast, DMA*-SH uses a hypernetwork h_η to generate adapter parameters $\omega = h_\eta(z_t)$ that modulate small bottleneck modules injected into the policy, action-value, and dynamics models (Figure 6). A single adapter with a residual (skip)

connection is inserted after the feature trunk and before the output head in each network. Let x_t denote the trunk features. The adapted features are:

$$\tilde{x}_t = x_t + g_{\text{adapter}}(x_t; \omega), \quad (3)$$

where g_{adapter} is a bottleneck network whose weights ω are generated by the hypernetwork. This design provides context-dependent modulation of features before they are mapped to action means, action-value estimates, or predicted state differences.

The modulation is multiplicative in the following sense. Inside the adapter, features are transformed by weight matrices that depend on context through $\omega = h_\eta(z_t)$. Concretely, the computation contains terms of the form $W(z_t)x_t$, which induces explicit multiplicative interactions between feature coordinates and context coordinates. This form of multiplicative modulation [Jayakumar et al., 2020] can exactly represent certain functions outside the concatenation hypothesis class.

Theorems A.1, A.2, and A.5 in Appendix A.1 formalize this separation by showing that hypernetwork-conditioned adapters can represent context-dependent multiplicative transformations outside the hypothesis class of concatenation-based ReLU policies. In non-overlapping context shifts, this capacity supports efficient representation of discontinuous changes in action effects, whereas concatenation-based methods must realize such structure only indirectly (Remark A.6).

5 The Actuator Inversion Benchmark (AIB)

We introduce the Actuator Inversion Benchmark (AIB) as a diagnostic suite for contextual RL with discontinuous context-to-dynamics interactions. The name reflects the minimal motivating case: actuator inversion, where the same motor command produces opposite effects under a latent mode. Beyond sign flips, AIB includes actuator permutations, where control channels are swapped, and continuous weakly non-overlapping systems such as ODE/ODE-k, where the required action direction changes across regions of the continuous context space. AIB therefore isolates when zero-shot contextual RL requires reliable mode identification and mode-conditioned computation, while still encompassing standard physical parameter variations that primarily demand smooth interpolation. Each AIB environment has two context dimensions and is classified as either:

- **Overlapping:** A single context-unaware policy can perform well across contexts, as with continuous physical parameters such as mass or gravity.
- **Non-overlapping:** Every context-unaware policy without memory incurs nontrivial regret in at least one context, requiring context conditioning.

Formal definitions appear in Appendix A.2 (Definition A.8). Table 1 summarizes all AIB environments, their context variables, and overlap type. Appendix E provides task descriptions and classification rationales, and Table 5 specifies context supports and return bounds for score normalization. Unless noted otherwise, each environment samples a two-dimensional context once per episode and holds it fixed throughout; context instances are pairwise disjoint across $\mathcal{C}_{\text{train}}$, $\mathcal{C}_{\text{eval-in}}$, and $\mathcal{C}_{\text{eval-out}}$.

Unlike CARL [Benjamins et al., 2023], which focuses on continuous and categorical context variations, or Meta-World [Yu et al., 2020], which varies goals and tasks but not action effects, AIB includes discontinuous changes in action effects, such as sign inversions and channel permutations, yielding mutually incompatible control laws. See Appendix E.2.

6 Results

6.1 Metrics

We adopt a standard evaluation protocol for zero-shot generalization in contextual RL [Kirk et al., 2023, Beukman et al., 2023, Benjamins et al., 2023]. Specifically, we sample $n_c = 20$ contexts from the environment-specific context ranges listed in Table 5 to create the sets $\mathcal{C}_{\text{train}}$, $\mathcal{C}_{\text{eval-in}}$, and $\mathcal{C}_{\text{eval-out}}$, respectively. The agent is trained on $\mathcal{C}_{\text{train}}$. For evaluation, we measure the cumulative episodic return of the trained agent across $n_e = 10$ rollouts per context, and then average within each context set. This yields three averaged episodic returns (AER) [Beukman et al., 2023], one per set. Following Agarwal et al. [2021], we report the interquartile mean (IQM) with empirical confidence intervals, after min-max scaling by environment-specific return bounds (Table 5; Appendix E). Unless stated otherwise, results are averaged over $n_s = 10$ independent random seeds.

Table 1: **Actuator Inversion Benchmark (AIB)**. Each environment has two context dimensions (except ODE- k , which uses k dimensions). *Non-overlapping* environments include either a binary actuator inversion factor $c \in \{\pm 1\}$ that flips action effects, or an actuator permutation factor $q \in \{0, 1\}$ that swaps action dimensions, both typically yielding mutually incompatible control laws. *Overlapping* environments vary only continuous physical parameters. *Weakly non-overlapping* denotes continuous contexts with empirically low policy overlap. [†]ODE/ODE- k adapted from prior work; details and attribution in Appendix E.

Environment	Source	Context variables	Type
DI	Custom	mass; actuator inversion (± 1)	Non-overlap
DI-Friction	Custom	mass; friction	Overlap
DI-Perm	Custom	mass; actuator permutation $q \in \{0, 1\}$	Non-overlap
ODE	Custom [†]	ODE coeffs (c_1, c_2)	Weakly non-overlap
ODE- k	Custom [†]	polynomial ODE coeffs (c_1, \dots, c_k)	Weakly non-overlap
Cartpole	DMC	pole length; actuator inversion (± 1)	Non-overlap
Cheetah	DMC	leg length; actuator inversion (± 1)	Non-overlap
Reacher (E/H)	DMC	arm length; actuator inversion (± 1)	Non-overlap
Reacher (E/H)-Perm	DMC	arm length; actuator permutation $q \in \{0, 1\}$	Non-overlap
BallInCup	DMC	tendon length; gravity	Overlap
Walker	DMC	actuator strength; gravity	Overlap
WalkerGym	Gymnasium	actuator strength; gravity	Overlap
HopperGym	Gymnasium	actuator strength; gravity	Overlap

6.2 Baselines

For our methods DMA* and DMA*-SH, as well as for all baselines except Amago, we use Soft Actor-Critic (SAC) [Haarnoja et al., 2018] as the underlying RL algorithm. To ensure comparability, we use SAC with standard hyperparameters and avoid additional tuning. Hyperparameters and implementation details are in Appendix D. All approaches are trained under the same procedure: the agent is trained in parallel across the $n_c = 20$ contexts in $\mathcal{C}_{\text{train}}$.

Concat (Context-Aware). This baseline concatenates explicit context with the state as input to the policy and Q-function, a standard approach when context variables are available [Ball et al., 2021, Eghbal-zadeh et al., 2021].

Decision Adapter (DA) (Context-Aware). Beukman et al. [2023] propose a stronger context-aware baseline that uses hypernetworks to adapt policy and Q-function parameters based on the context, improving over other context-aware approaches such as FLAP [Peng et al., 2021] and cGate [Benjamins et al., 2023].

Domain Randomization (DR) (Context-Unaware). This baseline ignores explicit context and relies on domain randomization [Tobin et al., 2017] across multiple contexts.

Amago (Context-Unaware). Recurrent agents can accumulate latent information over time, enabling in-context adaptation. Amago [Grigsby et al., 2024a] is a general-purpose in-context meta-RL algorithm. We use the improved Amago-2 variant [Grigsby et al., 2024b] with a GRU trajectory encoder (Figure 16). In Appendix A.6, we evaluate a Transformer variant to test whether history-based sequence modeling can replace DMA*-SH’s dynamics-grounded shared operator map.

Dynamic Model Alignment (DMA) (Context-Inferred). Prior methods such as DALI [Röder et al., 2025], IIDA [Evans et al., 2022], and CaDM [Lee et al., 2020] infer context from recent experience via dynamic model alignment. The inferred latent representation is then passed to the policy and Q-function. As DMA* extends this paradigm, we include vanilla DMA as a baseline.

DMA-Pearl (Context-Inferred). Pearl [Rakelly et al., 2019] is a meta-RL algorithm that infers context with a probabilistic encoder trained via Q-function gradients. While Rakelly et al. [2019] evaluated Pearl primarily under reward variations, we adapt it to transition dynamics variations by training the context encoder jointly with a dynamics model. This yields a probabilistic extension of DMA, where the context representation is regularized by a KL penalty against a unit Gaussian prior $\mathcal{N}(0, I)$, weighted by $\beta = 0.4$ (Figure 17).

6.3 Zero-Shot Generalization

IQM scores aggregated across all contextualized environments (Figure 2) show that DMA* and DMA*-SH achieve strong zero-shot generalization, particularly in the out-of-distribution setting.

The strongest competitors are the context-aware Concat and DA baselines, which DMA*-SH consistently outperforms across all three regimes. DMA*-SH also attains consistently strong AER scores across environments and contextualization types (Table 2). We complement these aggregated metrics with a per-context analysis, which reveals how performance changes as evaluation contexts diverge from the training distribution and exposes failure modes that aggregated statistics can obscure. Full per-context heatmaps, bar plots, and learning curves are provided in Appendix G.

In Walker, simple domain randomization suffices, suggesting that explicit or inferred context can sometimes hinder performance. Despite not being tailored to transition-dynamics variation, the context-unaware Amago algorithm performs competitively, especially in the training and eval-in regime, in most environments including non-overlapping settings such as DI, which cannot be solved by simple domain randomization (unlike DI-Friction with overlapping contexts). We note that Amago uses a substantially higher parameter count than the other SAC-based approaches in our experiments. DMA-Pearl achieves strong results in overlapping contextualizations; however, its smooth KL prior makes it uncompetitive in non-overlapping settings (Remark A.13 and Appendix F).

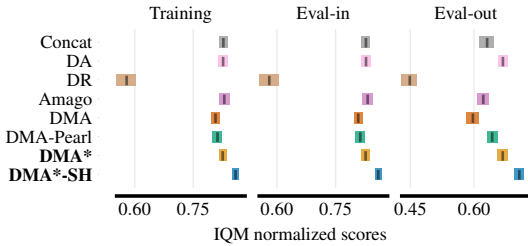


Figure 2: Interquartile mean (IQM) with 95% confidence intervals computed from AER scores, aggregated across environments. Results are reported for the three context sets $\mathcal{C}_{\text{train}}$, $\mathcal{C}_{\text{eval-in}}$, and $\mathcal{C}_{\text{eval-out}}$, comparing DMA*, DMA*-SH, and all baselines.

Table 2: AER scores with 95% confidence intervals for all environments. Results are aggregated across context sets $\mathcal{C}_{\text{train}}$, $\mathcal{C}_{\text{eval-in}}$, and $\mathcal{C}_{\text{eval-out}}$, comparing DMA*, DMA*-SH, and all baselines. Best AER scores are in bold; if multiple methods are highlighted for an environment, their differences are not distinguishable under the probability of improvement with 95% confidence intervals [Agarwal et al., 2021]. All aggregates all environments, *Overlap* aggregates those with overlapping contexts, and *Non-overlap* aggregates those with non-overlapping contexts. For the rows *All*, *Overlap*, and *Non-overlap*, scores are aggregated after environment-wise min-max scaling using environment-specific return bounds (Table 5). Per-set AER tables are provided in Appendix G.

Environment	Context-Aware		Context-Unaware				Context-Inferred		
	Concat	DA	DR	Amago	DMA	DMA-Pearl	DMA*	DMA*-SH	
DI	72 [69, 74]	75 [74, 76]	16 [9, 24]	61 [52, 70]	63 [62, 65]	68 [66, 70]	75 [74, 75]	76 [75, 77]	
DI-Friction	63 [50, 74]	76 [74, 77]	69 [53, 77]	79 [78, 79]	56 [41, 69]	74 [73, 76]	68 [52, 77]	77 [76, 77]	
DI-Perm	73 [71, 74]	76 [75, 77]	50 [40, 58]	72 [66, 77]	68 [66, 70]	70 [69, 71]	72 [70, 74]	75 [73, 76]	
ODE	162 [156, 167]	179 [176, 183]	63 [54, 72]	168 [167, 169]	166 [164, 168]	171 [165, 176]	175 [170, 180]	179 [176, 182]	
Cartpole	863 [851, 877]	892 [857, 926]	644 [595, 691]	639 [568, 715]	900 [879, 923]	884 [839, 917]	927 [904, 950]	967 [954, 978]	
Cheetah	385 [372, 402]	384 [373, 396]	281 [261, 303]	414 [402, 429]	388 [366, 411]	355 [335, 379]	383 [368, 399]	408 [390, 425]	
Reacher (E)	890 [859, 922]	878 [843, 910]	564 [515, 614]	914 [901, 925]	871 [834, 904]	875 [845, 905]	894 [870, 918]	898 [881, 916]	
Reacher (E)-Perm	883 [853, 910]	866 [825, 899]	659 [625, 701]	862 [844, 883]	881 [849, 912]	892 [870, 911]	869 [841, 897]	903 [881, 922]	
Reacher (H)	683 [569, 774]	713 [668, 763]	266 [185, 350]	853 [831, 872]	654 [573, 734]	686 [606, 746]	697 [611, 771]	805 [743, 855]	
Reacher (H)-Perm	712 [624, 789]	749 [711, 790]	328 [244, 417]	739 [693, 779]	668 [558, 751]	668 [609, 730]	679 [576, 764]	840 [800, 873]	
BallInCup	924 [915, 932]	879 [868, 892]	862 [824, 887]	651 [540, 763]	912 [904, 920]	903 [895, 912]	900 [884, 914]	890 [878, 902]	
Walker	783 [772, 794]	784 [765, 801]	798 [785, 810]	753 [740, 764]	762 [718, 790]	804 [796, 811]	780 [763, 793]	806 [798, 814]	
WalkerGym	2701 [2418, 2960]	3321 [3180, 3486]	2811 [2626, 2997]	3924 [3675, 4207]	3102 [2919, 3307]	3162 [2902, 3417]	2960 [2652, 3264]	3258 [3170, 3338]	
HopperGym	2521 [2434, 2606]	2542 [2485, 2601]	2377 [2313, 2437]	2501 [2427, 2569]	2629 [2549, 2696]	2646 [2595, 2696]	2635 [2586, 2692]	2563 [2535, 2588]	
All	0.73 [0.71, 0.75]	0.76 [0.75, 0.77]	0.52 [0.5, 0.54]	0.73 [0.72, 0.74]	0.72 [0.7, 0.74]	0.74 [0.73, 0.75]	0.75 [0.73, 0.76]	0.79 [0.78, 0.79]	
Overlap	0.71 [0.68, 0.74]	0.75 [0.74, 0.76]	0.71 [0.67, 0.73]	0.73 [0.7, 0.75]	0.71 [0.68, 0.74]	0.76 [0.74, 0.77]	0.73 [0.69, 0.75]	0.76 [0.75, 0.76]	
Non-overlap	0.74 [0.72, 0.76]	0.77 [0.75, 0.78]	0.41 [0.39, 0.44]	0.73 [0.72, 0.75]	0.72 [0.7, 0.74]	0.73 [0.72, 0.74]	0.75 [0.74, 0.77]	0.8 [0.79, 0.81]	

6.4 Context-Embedding Diagnostics and Geometry

To analyze how RL task performance relates to z_t , we introduce three evaluation criteria: *Informativeness*, *Variability*, and *Representation-Overlap* (RO).

For each split $M \in \{\text{train}, \text{eval-in}, \text{eval-out}\}$, we collect a dataset of trajectory windows τ from contexts $c \in \mathcal{C}_M$ and embed each window as $z = g_\phi(\tau)$.

Informativeness. We quantify how strongly the embedding z_t depends on the true context c via the mutual information $I(z_t; c)$. We estimate $I(z_t; c)$ using the k -nearest-neighbors entropy estimator

[Kraskov et al., 2004] with $k = 3$ under L_2 distance, and use this estimate as a diagnostic of context dependence in the learned representation [Garcin et al., 2025]. A higher value of $I(z_t; c)$ signifies that z_t varies more systematically with c across the collected trajectories.

Variability. We measure dataset-level spread of context representations $z \in \mathbb{R}^{d_z}$ as $\text{Variability}(M) = \frac{1}{d_z} \text{tr}(\text{Cov}(Z))$ (Definition A.14 in Appendix A.3). Lower Variability corresponds to more consistent context signals across trajectory windows, which can facilitate robust policy learning.

Representation-Overlap (RO). RO is the average pairwise cosine similarity between per-context mean embeddings over all context pairs (Definition A.21, (30)). Higher RO indicates stronger global directional concentration of context means in representation space.

Figure 3 shows that Variability decreases along the ablation path $\text{DMA} \rightarrow \text{DMA}^* \rightarrow \text{DMA}^*\text{-SH}$ across tasks. Theorem A.17 (Appendix A.3.1) shows lower Variability tightens gradient-variance bounds, explaining this correlation with RL performance. Surprisingly, $\text{DMA}^*\text{-SH}$ has lower Informativeness $I(z_t; c)$ yet better returns. Proposition A.18 resolves this apparent paradox: within-mode compression reduces policy-gradient variance even as total information decreases. Interpreted through an approximate *structural information bottleneck* lens (Appendix A.3.4), this reflects mode sufficiency with selective within-mode compression. The key structural link is Theorem A.15, which decomposes Variability into within-context noise, within-mode spread along continuous context dimensions, and between-mode separation induced by actuator inversion.

These findings are complemented by a RO analysis (Figure 3). Across our ablations, $\text{DMA}^*\text{-SH}$ attains higher RO, suggesting more *directionally concentrated* representations: within-mode variation induces smaller directional changes, while actuator-inversion modes remain separated. In DI, the t-SNE and cosine analyses in Figure 4 are consistent with reduced within-mode spread across mass while preserving separation across actuator inversion, aligning with the compression and separation terms in Theorem A.15. See Appendix A.3 for details on directional geometry induced by normalization and shared hypernetwork conditioning.

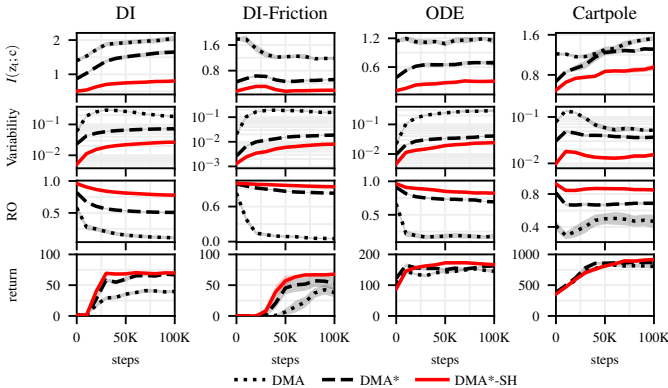


Figure 3: Informativeness $I(z_t; c)$, Variability, Representation-Overlap (RO), and episodic returns on $\mathcal{C}_{\text{eval-out}}$. $\text{DMA}^*\text{-SH}$ reduces Variability and increases RO, indicating directionally concentrated embeddings that are less sensitive to within-mode nuisance variation while preserving task-relevant mode structure. This geometry correlates with stronger zero-shot returns and supports the structural information-bottleneck interpretation in Appendix A.3.4.

6.5 Implicit Gradient Regularization via Shared Hypernetworks

In $\text{DMA}^*\text{-SH}$, shared adapters $\omega = h_\eta(z_t)$ trained only by (2) and detached in RL losses can be viewed as an implicit gradient regularizer for context-conditioned policy learning. We compare $\text{DMA}^*\text{-SH}$ to a separate-hypernetwork variant, $\text{DMA}^*\text{-H}$, in which the dynamics, actor, and critic each have their own hypernetwork (with parameters η^f, η^π, η^Q , producing $\omega^f, \omega^\pi, \omega^Q$), and the actor/critic hypernetworks can adapt directly to their respective RL objectives. To quantify how strongly the policy objective depends on the inferred context through the adapter pathway, we report the mean norm of a *shadow* z -space gradient, $\mathbb{E}\|\nabla_z L_\pi\|$, in Figure 5. The shadow gradient is computed by temporarily removing the stop-gradient on $\omega = h_\eta(z)$; under the actual training rule in $\text{DMA}^*\text{-SH}$, $\nabla_z L_\pi = 0$ since ω is detached in the RL losses and z_t is used by the actor only through ω . Shadow gradients serve purely as diagnostic signatures of effective context utilization; they are never applied during training. Across environments, $\text{DMA}^*\text{-SH}$ shows persistently non-negligible shadow norms, indicating sustained hypothetical sensitivity of the policy objective to the context signal under shared, dynamics-trained adapters. In contrast, $\text{DMA}^*\text{-H}$ exhibits substantially smaller

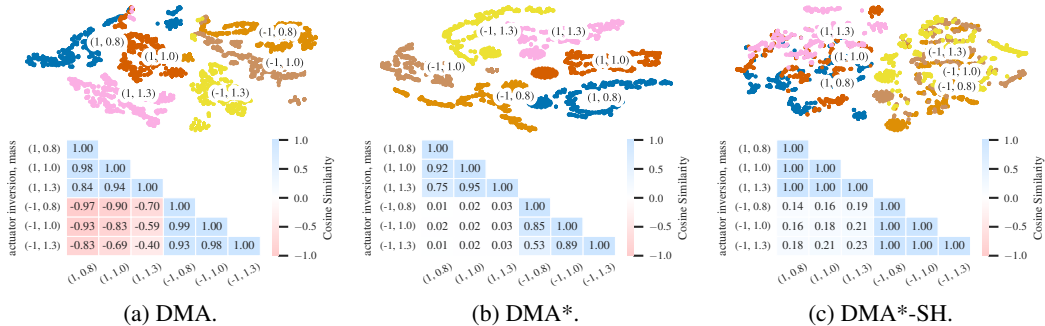


Figure 4: DI (non-overlapping): t-SNE of inferred embeddings z_t (top) and cosine similarity heatmaps of per-context mean embeddings (bottom; (29)) for DMA, DMA*, and DMA*-SH. DMA*-SH shows stronger within-mode alignment across the continuous mass dimension while maintaining separation between actuator-inversion modes, consistent with the compression/separation terms in Theorem A.15. Mass clusters overlap more for DMA*-SH, yet returns are higher, consistent with mass having largely overlapping policy effects.

values, consistent with weaker effective context dependence along the corresponding adapter pathway. This separation co-occurs with faster learning and higher returns for DMA*-SH (Figure 5). See Appendices A.5 and A.6 for an extended discussion and additional diagnostics.

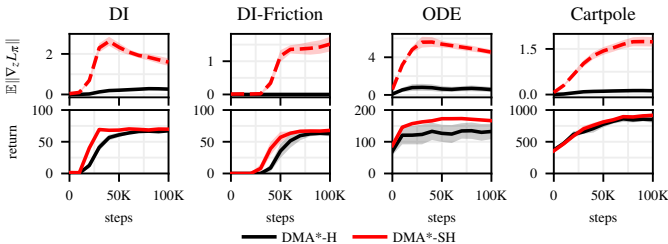


Figure 5: Implicit regularization of RL via a dynamics-trained shared hypernetwork. Top: Mean policy context sensitivity in shared embedding space $\mathbb{E}\|\nabla_z L_\pi\|$. Bottom: Episodic returns.

7 Conclusions

We introduced the Actuator Inversion Benchmark (AIB) as a diagnostic suite for zero-shot generalization under discontinuous context-to-dynamics shifts. We proposed DMA*-SH, a context-inferred RL framework using a shared hypernetwork to couple latent context inference to dynamics-aligned modulation. Theoretically, we proved expressivity separation results for hypernetwork-conditioned adapters (Theorems A.1, A.2, A.5), identifying multiplicative modulation as an appropriate inductive primitive for non-overlapping context shifts. Empirically, DMA*-SH embeddings show stronger within-mode directional alignment across continuous context variation while preserving mode separation. Theorem A.15 connects these patterns to Variability through within-mode compression and between-mode separation; Theorem A.17 links reduced Variability to tighter policy-gradient variance bounds. Together, these results support an approximate structural information bottleneck view: preserve mode-relevant information while compressing nuisance variation. Finally, shared hypernetwork design regularizes RL updates toward dynamics-consistent solutions. Notably, these effects emerge *without* auxiliary objectives. Directional concentration, nuisance compression, structural-bottleneck-like behavior, and implicit gradient regularization arise from the interlocking design of dynamics-aligned training, shared modulation, and input/output normalization. Overall, our results support DMA*-SH as a principled approach to zero-shot contextual RL under discontinuous shifts.

Limitations and Future Work. DMA*-SH couples context inference to learned dynamics through a shared hypernetwork, so model errors can propagate into the representation and impair adaptation under misspecification or rapidly shifting dynamics. Multiplicative modulation is natural for discontinuous action-effect shifts, but may be less effective when context modifies rewards or induces non-factorizable policy changes. Hypernetwork capacity is also a practical bottleneck: too little limits expressiveness, while too much can overfit and reduce stability.

Promising directions include robustness to model uncertainty via ensembles or Bayesian hypernetworks [Krueger et al., 2017], extensions to reward-modifying contexts via dual hypernetworks or multi-view encoders, and explicit objectives for strengthening compression/separation [Li et al., 2024a,b]. Real-robot deployments involving actuator degradation, swapped channels, or intermittent faults provide natural testbeds for shared modulation under genuinely discontinuous dynamics shifts.

Acknowledgments and Disclosure of Funding

JB and ME gratefully acknowledge funding by the German Research Foundation DFG through the MoReSpace (402776968) project.

References

- Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, 2021.
- Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- Philip J Ball, Cong Lu, Jack Parker-Holder, and Stephen Roberts. Augmented world models facilitate zero-shot dynamics generalization from a single offline environment. In *International Conference on Machine Learning*, 2021.
- Pradeep Kr. Banerjee and Guido Montúfar. The variational deficiency bottleneck. In *International Joint Conference on Neural Networks*, 2020.
- Jacob Beck, Matthew Jackson, Risto Vuorio, and Shimon Whiteson. Hypernetworks in meta-reinforcement learning. In *Conference on Robot Learning*, 2022.
- Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*, 2023a.
- Jacob Beck, Risto Vuorio, Zheng Xiong, and Shimon Whiteson. Recurrent hypernetworks are surprisingly strong in meta-rl. In *Advances in Neural Information Processing Systems*, 2023b.
- Carolin Benjamins, Theresa Eimer, Frederik Schubert, Aditya Mohan, Sebastian Döhler, André Biedenkapp, Bodo Rosenhahn, Frank Hutter, and Marius Lindauer. Contextualize me – the case for context in reinforcement learning. *Transactions on Machine Learning Research*, 2023.
- Michael Beukman, Devon Jarvis, Richard Klein, Steven James, and Benjamin Rosman. Dynamics generalisation in reinforcement learning via adaptive context-aware policies. In *Advances in Neural Information Processing Systems*, 2023.
- Martin V. Butz. Towards a unified sub-symbolic computational theory of cognition. *Frontiers in Psychology*, 2016.
- Martin V. Butz. Resourceful event-predictive inference: The nature of cognitive effort. *Frontiers in Psychology*, 2022.
- Martin V. Butz, David Bilkey, Dania Humaidan, Alistair Knott, and Sebastian Otte. Learning, planning, and control in a monolithic neural event inference architecture. *Neural Networks*, 2019.
- Martin V. Butz, Maximilian Mittenbühler, Sarah Schwöbel, Asya Achimova, Christian Gumbsch, Sebastian Otte, and Stefan Kiebel. Contextualizing predictive minds. *Neuroscience & Biobehavioral Reviews*, 2024.
- Tao Chen, Adithyavairavan Murali, and Abhinav Gupta. Hardware conditioned policies for multi-robot transfer learning. In *Advances in Neural Information Processing Systems*, 2018.
- Dario Cuevas Rivera and Stefan Kiebel. The effects of probabilistic context inference on motor adaptation. *PLOS ONE*, 2023.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.
- Ron Dorfman, Idan Shenfeld, and Aviv Tamar. Offline meta reinforcement learning—identifiability challenges and effective data collection strategies. In *Advances in Neural Information Processing Systems*, 2021.
- Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. RL²: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.
- Hamid Eghbal-zadeh, Florian Henkel, and Gerhard Widmer. Context-adaptive reinforcement learning using unsupervised learning of context variables. In *NeurIPS Workshop on Pre-registration in Machine Learning*, 2021.
- Ben Evans, Abitha Thankaraj, and Lerrel Pinto. Context is everything: Implicit identification for dynamics adaptation. In *International Conference on Robotics and Automation*, 2022.
- Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.
- Scott Fujimoto, Wei-Di Chang, Edward Smith, Shixiang Shane Gu, Doina Precup, and David Meger. For sale: State-action representation learning for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.
- Tomer Galanti and Lior Wolf. On the modularity of hypernetworks. In *Advances in Neural Information Processing Systems*, 2020.
- Samuel Garcin, Trevor McInroe, Pablo Samuel Castro, Christopher G. Lucas, David Abel, Prakash Panangaden, and Stefano V Albrecht. Studying the interplay between the actor and critic representations in reinforcement learning. In *International Conference on Learning Representations*, 2025.
- Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, 2019.
- Dibya Ghosh, Jad Rahme, Aviral Kumar, Amy Zhang, Ryan P Adams, and Sergey Levine. Why generalization in RL is difficult: Epistemic POMDPs and implicit partial observability. In *Advances in Neural Information Processing Systems*, 2021.
- Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J. Cree. Regularisation of neural networks by enforcing Lipschitz continuity. *Machine Learning*, 2021.
- Jake Grigsby, Linxi Fan, and Yuke Zhu. AMAGO: Scalable in-context reinforcement learning for adaptive agents. In *International Conference on Learning Representations*, 2024a.
- Jake Grigsby, Justin Sasek, Samyak Parajuli, Daniel Adeb, Amy Zhang, and Yuke Zhu. AMAGO-2: Breaking the multi-task barrier in meta-reinforcement learning with transformers. In *Advances in Neural Information Processing Systems*, 2024b.
- David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017.
- Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 2018.
- Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019.
- Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 2025.
- Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual Markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.
- Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.
- Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Conference on Computer Vision and Pattern Recognition*, 2022.

- James B. Heald, Máté Lengyel, and Daniel M. Wolpert. Contextual inference underlies the learning of sensorimotor repertoires. *Nature*, 2021.
- James B. Heald, Máté Lengyel, and Daniel M. Wolpert. Contextual inference in learning and memory. *Trends in Cognitive Sciences*, 2023.
- Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. CleanRL: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 2022.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.
- Siddhant M Jayakumar, Wojciech M Czarnecki, Jacob Menick, Jonathan Schwarz, Jack Rae, Simon Osindero, Yee Whye Teh, Tim Harley, and Razvan Pascanu. Multiplicative interactions and where to find them. In *International Conference on Learning Representations*, 2020.
- Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 2023.
- Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E – Statistical, Nonlinear, and Soft Matter Physics*, 2004.
- David Krueger, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron Courville. Bayesian hypernetworks. *arXiv preprint arXiv:1710.04759*, 2017.
- Samuel Lavoie, Christos Tsirigotis, Max Schwarzer, Ankit Vani, Michael Noukhovitch, Kenji Kawaguchi, and Aaron Courville. Simplicial embeddings in self-supervised learning and downstream classification. In *International Conference on Learning Representations*, 2023.
- Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In *International Conference on Machine Learning*, 2020.
- Lanqing Li, Rui Yang, and Dijun Luo. FOCAL: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. In *International Conference on Learning Representations*, 2021.
- Lanqing Li, Hai Zhang, Xinyu Zhang, Shatong Zhu, Yang Yu, Junqiao Zhao, and Pheng-Ann Heng. Towards an information theoretic framework of context-based offline meta-reinforcement learning. In *Advances in Neural Information Processing Systems*, 2024a.
- Zhengwei Li, Zhenyang Lin, Yurou Chen, and Zhiyong Liu. Efficient offline meta-reinforcement learning via robust task representations and adaptive policy generation. In *International Joint Conference on Artificial Intelligence*, 2024b.
- Fangchen Liu, Hao Liu, Aditya Grover, and Pieter Abbeel. Masked autoencoding for scalable and generalizable decision making. In *Advances in Neural Information Processing Systems*, 2022.
- Fan-Ming Luo, Zuolin Tu, Zefang Huang, and Yang Yu. Efficient recurrent off-policy rl requires a context-encoder-specific learning rate. In *Advances in Neural Information Processing Systems*, 2024.
- Dimitrije Marković, Thomas Goschke, and Stefan J. Kiebel. Meta-control of the exploration-exploitation dilemma emerges from probabilistic inference over a hierarchy of time scales. *Cognitive, Affective, & Behavioral Neuroscience*, 2021.
- Reginald McLean, Evangelos Chatzaroulas, Luc McCutcheon, Frank Röder, Tianhe Yu, Zhanpeng He, K.R. Zentner, Ryan Julian, J K Terry, Isaac Woungang, Nariman Farsad, and Pablo Samuel Castro. Meta-world+: An improved, standardized, RL benchmark. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.
- Luckeciano C Melo. Transformers are meta-reinforcement learners. In *International Conference on Machine Learning*, 2022.
- Maximilian Mittenbühler, Sarah Schwöbel, David Dignath, Stefan Kiebel, and Martin Butz. A rational trade-off between the costs and benefits of automatic and controlled processing. In *Cognitive Science Conference*, 2024.
- Aditya Modi, Nan Jiang, Satinder Singh, and Ambuj Tewari. Markov decision processes with continuous side information. In *Algorithmic Learning Theory*, 2018.

- Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, 2014.
- Yao Mu, Yuzheng Zhuang, Fei Ni, Bin Wang, Jianyu Chen, Jianye Hao, and Ping Luo. Domino: Decomposed mutual information optimization for generalized context in meta-reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.
- Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2018.
- Tidiane Camaret Ndir, André Biedenkapp, and Noor Awad. Inferring behavior-specific context improves zero-shot generalization in reinforcement learning. In *European Workshop on Reinforcement Learning*, 2024.
- Johan Obando-Ceron, Walter Mayor, Samuel Lavoie, Scott Fujimoto, Aaron Courville, and Pablo Samuel Castro. Simplicial embeddings improve sample efficiency in actor-critic agents. *arXiv preprint arXiv:2510.13704*, 2025.
- Thomas Parr, Emma Holmes, Karl J. Friston, and Giovanni Pezzulo. Cognitive effort and active inference. *Neuropsychologia*, 2023.
- Matt Peng, Banghua Zhu, and Jiantao Jiao. Linear representation meta-reinforcement learning for instant adaptation. *arXiv preprint arXiv:2101.04750*, 2021.
- Jordan B Pollack. Recursive distributed representations. *Artificial Intelligence*, 1990.
- Sai Prasanna, Karim Farid, Raghu Rajan, and André Biedenkapp. Dreaming of many worlds: Learning contextual world models aids zero-shot generalization. *Reinforcement Learning Journal*, 2024.
- Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning*, 2019.
- Frank Röder, Jan Benad, Manfred Eppe, and Pradeep Kr. Banerjee. Dynamics-aligned latent imagination in contextual world models for zero-shot generalization. In *Advances in Neural Information Processing Systems*, 2025.
- Fedor Scholz, Christian Gumbsch, Sebastian Otte, and Martin V. Butz. Inference of affordances and active motor control in simulated agents. *Frontiers in Neurobotics*, 2022.
- Sarah Schwöbel, Dimitrije Marković, Michael N. Smolka, and Stefan J. Kiebel. Balancing control: A Bayesian interpretation of habitual and goal-directed behavior. *Journal of Mathematical Psychology*, 2021.
- Younggyo Seo, Kimin Lee, Ignasi Clavera Gilaberte, Thanard Kurutach, Jinwoo Shin, and Pieter Abbeel. Trajectory-wise multiple choice learning for dynamics generalization in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.
- Seyed Kamyar Seyed Ghasemipour, Shixiang Shane Gu, and Richard Zemel. Smile: Scalable meta inverse reinforcement learning through context-conditional policies. In *Advances in Neural Information Processing Systems*, 2019.
- Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. In *International Conference on Machine Learning*, 2021.
- Shagun Sodhani, Franziska Meier, Joelle Pineau, and Amy Zhang. Block contextual MDPs for continual learning. In *Learning for Dynamics and Control Conference*, 2022.
- Yuuya Sugita, Jun Tani, and Martin V Butz. Simultaneously emerging braitenberg codes and compositionality. *Adaptive Behavior*, 2011.
- Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. DeepMind Control Suite. *arXiv preprint arXiv:1801.00690*, 2018.
- Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.
- Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *International Conference on Intelligent Robots and Systems*, 2017.

- Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulao, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.
- Manuel Traub, Frederic Becker, Adrian Sauter, Sebastian Otte, and Martin V. Butz. Loci-segmented: Improving scene segmentation learning. In *International Conference on Artificial Neural Networks*, 2024.
- Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008.
- Johannes von Oswald, Christian Henning, Benjamin F. Grewe, and João Sacramento. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2020.
- Zhou Xian, Shamit Lal, Hsiao-Yu Tung, Emmanouil Antonios Platanios, and Katerina Fragkiadaki. Hyperdynamics: Meta-learning object and agent dynamics with hypernetworks. In *International Conference on Learning Representations*, 2021.
- Zhenjia Xu, Jiajun Wu, Andy Zeng, Joshua B Tenenbaum, and Shuran Song. Densephysnet: Learning dense physical object representations via multi-step dynamic interactions. In *Robotics: Science and Systems*, 2019.
- Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, 2020.
- Haoqi Yuan and Zongqing Lu. Robust task representations for offline meta-reinforcement learning via contrastive learning. In *International Conference on Machine Learning*, 2022.
- Jeffrey M. Zacks and Barbara Tversky. Event structure in perception and conception. *Psychological Bulletin*, 2001.
- Jeffrey M. Zacks, Nicole K. Speer, Khena M. Swallow, Todd S. Braver, and Jeremy R. Reynolds. Event perception: A mind-brain perspective. *Psychological Bulletin*, 2007.
- Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, 2019.
- Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. VariBAD: A very good method for Bayes-adaptive deep RL via meta-learning. In *International Conference on Learning Representation*, 2020.

Appendix

A	Theoretical Results and Supplementary Analyses	16
A.1	Multiplicative Interactions in Contextual Policies	16
A.2	Policy Overlap: Overlapping and Non-Overlapping Contexts	21
A.2.1	Context-Aware (e.g., Concat/DA Baselines)	22
A.2.2	Context-Unaware (e.g., Standard DR with Markovian Policy)	22
A.2.3	Context-Inferred (Our Method, DMA*-SH)	23
A.3	Informativeness, Variability, and Gradient Variance	24
A.3.1	Decomposing Variability into Noise, Compression, and Separation	24
A.3.2	Variability controls policy-gradient variance	26
A.3.3	The Informativeness–Variability paradox: When less information improves learning	28
A.3.4	DMA*-SH as an Implicit Structural Information Bottleneck	29
A.3.5	Regime shifts and selective-compression signatures on $\mathcal{C}_{\text{eval-out}}$	32
A.4	Scale Control and Directional Geometry in the Context Space	34
A.4.1	Representation-Overlap (RO).	34
A.4.2	t-SNE Visualization and RO Cosine Similarity Analysis	35
A.5	Implicit Regularization via Shared Hypernetwork Gradients	35
A.6	Comparison to History-Based Recurrent and Transformer Agents	37
B	Extended Related Work	40
C	Algorithms	41
D	Hyperparameters and Implementation Details	42
E	AIB: Discontinuous Context-to-Dynamics Environment Suite	44
E.1	Environments	44
E.2	Comparison with Existing Benchmarks	46
F	Ablations and Design Rationale	47
G	Detailed Results	52
G.1	Separate Result Tables, Learning Curves and Performance Gains	52
G.2	Context-Instance Generalization Analysis	55
G.3	Scalability in the Explicit Context Dimension	62

A Theoretical Results and Supplementary Analyses

A.1 Multiplicative Interactions in Contextual Policies

One standard approach to conditioning a policy on a context embedding $z_t \in \mathbb{R}^{d_z}$ is through concatenation with the state s_t , yielding an input $[s_t; z_t]$ to a ReLU MLP. Such networks are continuous piecewise-linear (CPWL) in the joint input, with the Hessian matrix vanishing almost everywhere. CPWL functions can approximate complex interactions by using many linear regions, but they cannot represent truly bilinear couplings exactly on any domain with non-empty interior. As a result, mode-switching and sign-dependent transformations, such as those required for actuator inversion (Section 5), are typically realized only indirectly through a fine partition of the input space.

In contrast, *multiplicative interactions* enable richer structure via bilinear forms: $f(s, z) = z^\top W s +$ lower-order terms, where W captures cross-terms between s and z [Jayakumar et al., 2020, Galanti and Wolf, 2020]. Hypernetworks instantiate such interactions: when $h_\eta(z)$ generates weights for an adapter layer computing $W^{(\omega)}x$, where x are features derived from s , the map is piecewise bilinear in (x, z) , with regions determined by the activation patterns of both the hypernetwork and the adapter.

We show that hypernetwork-conditioned adapters can realize functions that concatenation ReLU MLPs cannot represent exactly. We first state the separation in the scalar-output and linear adapter case:

Theorem A.1 (Separation of hypernetwork-adapter and concatenation hypothesis classes). *Let $\mathcal{H}_{\text{concat}}$ be the class of functions $f : \mathbb{R}^{d_s} \times \mathbb{R}^{d_z} \rightarrow \mathbb{R}$ realized by finite ReLU MLPs on input $[s; z]$. Let $\mathcal{H}_{\text{hyper}}$ be the class of functions of the form*

$$f(s, z) = w^\top \left(x(s) + g_{\text{adapter}}(x(s); \omega = h_\eta(z)) \right) + b, \quad (4)$$

where:

- $x : \mathbb{R}^{d_s} \rightarrow \mathbb{R}^n$ is a finite ReLU network (trunk),
- $h_\eta : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_\omega}$ is a finite ReLU network with linear output layer (hypernetwork),
- for each $\omega \in \mathbb{R}^{d_\omega}$, $g_{\text{adapter}}(\cdot; \omega) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the linear map $x \mapsto W^{(\omega)}x$, where $W^{(\omega)} \in \mathbb{R}^{n \times n}$ is the adapter weight matrix parameterized by ω ,
- $w \in \mathbb{R}^n$, $b \in \mathbb{R}$ (linear output head).

Assume $s \in \mathcal{S} \subset \mathbb{R}^{d_s}$ and $z \in \mathcal{Z} \subset \mathbb{R}^{d_z}$ range over compact sets with non-empty interior. Then:

$$\mathcal{H}_{\text{hyper}} \not\subseteq \mathcal{H}_{\text{concat}}.$$

That is, there exists $f^* \in \mathcal{H}_{\text{hyper}}$ such that $f^* \notin \mathcal{H}_{\text{concat}}$.

Proof. We exhibit $f^*(s, z) = s \cdot z$ (scalar case, $d_s = d_z = 1$) and show $f^* \in \mathcal{H}_{\text{hyper}} \setminus \mathcal{H}_{\text{concat}}$.

Membership in $\mathcal{H}_{\text{hyper}}$. Choose $n = 2$. Define:

$$\begin{aligned} x(s) &= [\text{ReLU}(s), \text{ReLU}(-s)]^\top \in \mathbb{R}^2, \\ h_\eta(z) &= [z - 1, -(z + 1)]^\top \in \mathbb{R}^2, \\ W^{(\omega)} &= \text{diag}(\omega) = \begin{pmatrix} \omega_1 & 0 \\ 0 & \omega_2 \end{pmatrix}, \\ w &= [1, 1]^\top, \quad b = 0. \end{aligned}$$

The feature trunk x is a valid finite ReLU network. The hypernetwork h_η is affine in z and thus realizable by a ReLU network with linear output. The adapter computes $g_{\text{adapter}}(x; \omega) = W^{(\omega)}x = [\omega_1 x_1, \omega_2 x_2]^\top$, resulting in adapted features with a skip connection $\tilde{x} = x + g_{\text{adapter}}(x; \omega) = [(1 + \omega_1)x_1, (1 + \omega_2)x_2]^\top$. By substituting $\omega_1 = z - 1$ and $\omega_2 = -(z + 1)$, we obtain $\tilde{x} = [z \cdot \text{ReLU}(s), -z \cdot \text{ReLU}(-s)]^\top$. The output is

$$f(s, z) = w^\top \tilde{x} = z \cdot \text{ReLU}(s) - z \cdot \text{ReLU}(-s) = z \cdot s,$$

using the identity $\text{ReLU}(s) - \text{ReLU}(-s) = s$. Hence $f^* \in \mathcal{H}_{\text{hyper}}$.

Non-membership in $\mathcal{H}_{\text{concat}}$. Any finite ReLU MLP realizes a CPWL function [Montúfar et al., 2014]. A CPWL function is affine on each piece of a polyhedral partition, so its Hessian vanishes almost everywhere. However, $f^*(s, z) = sz$ has mixed partial derivative $\frac{\partial^2 f^*}{\partial s \partial z} = 1 \neq 0$ everywhere. No CPWL function can equal f^* on a domain with non-empty interior. Hence $f^* \notin \mathcal{H}_{\text{concat}}$. \square

Theorem A.1 proves the expressive gap in the minimal sign-flip setting. The following proposition and corollary show that the same hypernetwork-adapter mechanism extends to broader context-dependent operator switching. Theorem A.2 gives exact realization for broader operator families, including discrete permutations and gain modulation; Corollary A.3 gives an exact CPWL separation result for continuous affine operator families.

Theorem A.2 (Operator-family realization by hypernetwork-conditioned adapters). *Fix a feature map $x : \mathbb{R}^{d_s} \rightarrow \mathbb{R}^n$, an output vector $w \in \mathbb{R}^n$, a bias $b \in \mathbb{R}$, and a context domain $Z \subseteq \mathbb{R}^{d_z}$.*

Let $T : Z \rightarrow \mathbb{R}^{n \times n}$ be a matrix-valued map. Assume the hypernetwork and adapter parameterization can realize

$$W(h_\eta(z)) = T(z) - I_n \quad \text{for all } z \in Z, \quad (5)$$

where the linear adapter is $g_{\text{adapter}}(x; \omega) = W(\omega)x$. Define

$$f_T(s, z) = w^\top T(z)x(s) + b. \quad (6)$$

Then $f_T \in \mathcal{H}_{\text{hyper}}$. In particular:

1. *Actuator inversion: when the mode variable is a discrete input $c \in \{\pm 1\}$ (or an equivalent one-hot encoding), $T(c) \in \{I_n, -I_n\}$ gives $W \in \{0_{n \times n}, -2I_n\}$. The map $c \mapsto T(c) - I_n$ takes only two values on the discrete mode domain $c \in \{\pm 1\}$ and is therefore exactly realizable.*
2. *Permutation-induced wire swaps: when the mode is a discrete variable $q \in \{1, \dots, |G|\}$ indexing a finite group G , $T(q) = P_q$ is permutation-valued, and $W(h_\eta(q)) = P_q - I_n$ takes finitely many values. When q is presented as a discrete input (e.g., one-hot), the hypernetwork need only map finitely many input symbols to finitely many fixed output vectors, which is exactly realizable by a ReLU network with linear output.*
3. *Gain modulation corresponds to $T(z) = D(z)$ diagonal, and is exactly realizable when the diagonal entries are continuous piecewise-linear functions of z .*

Proof. By the definition of $\mathcal{H}_{\text{hyper}}$ in Theorem A.1, any function of the form

$$f(s, z) = w^\top (x(s) + g_{\text{adapter}}(x(s); h_\eta(z))) + b = w^\top (x(s) + W(h_\eta(z))x(s)) + b$$

belongs to $\mathcal{H}_{\text{hyper}}$. Using assumption (5),

$$\begin{aligned} x(s) + g_{\text{adapter}}(x(s); h_\eta(z)) &= x(s) + W(h_\eta(z))x(s) \\ &= x(s) + (T(z) - I_n)x(s) \\ &= T(z)x(s). \end{aligned} \quad (7)$$

Therefore $f(s, z) = w^\top T(z)x(s) + b = f_T(s, z)$, proving $f_T \in \mathcal{H}_{\text{hyper}}$.

For the specific cases:

1. $T(c) \in \{I_n, -I_n\}$ gives $W(h_\eta(c)) \in \{0_{n \times n}, -2I_n\}$. Since the mode variable c takes only two discrete values, the hypernetwork need only map two input symbols to two fixed output vectors, which is exactly realizable.
2. For a finite group G with $|G| < \infty$ and a discrete mode variable $q \in \{1, \dots, |G|\}$, $T(q) = P_q$ requires h_η to map q to the vectorized entries of $P_q - I_n$, a function taking finitely many values on a discrete domain. When q is presented as a discrete input (e.g., integer or one-hot), a ReLU network with linear output can realize this map exactly.
3. For $T(z) = \text{diag}(d_1(z), \dots, d_n(z))$ with entries that are continuous piecewise-linear in z , the map $z \mapsto T(z) - I_n$ is also CPWL and hence realizable by a finite ReLU network. \square

Corollary A.3 (Continuous operator-family separation). *Let $S \subset \mathbb{R}^{d_s}$ and $Z \subset \mathbb{R}^{d_z}$ be compact domains with non-empty interior. Suppose there exists a non-empty open set $U \subset S$ on which the feature map is affine:*

$$x(s) = As + a, \quad s \in U, \quad (8)$$

for some matrix $A \in \mathbb{R}^{n \times d_s}$ and vector $a \in \mathbb{R}^n$. Let

$$T(z) = T_0 + \sum_{j=1}^{d_z} z_j T_j, \quad z \in Z, \quad (9)$$

with fixed matrices $T_0, T_1, \dots, T_{d_z} \in \mathbb{R}^{n \times n}$, and define

$$f(s, z) = w^\top T(z) x(s) + b. \quad (10)$$

Assume there exist indices $i \in \{1, \dots, d_s\}$ and $j \in \{1, \dots, d_z\}$ such that

$$w^\top T_j A e_i \neq 0, \quad (11)$$

where e_i is the i -th standard basis vector in \mathbb{R}^{d_s} . Then

$$f \in \mathcal{H}_{\text{hyper}} \quad \text{and} \quad f \notin \mathcal{H}_{\text{concat}}.$$

Proof. Membership in $\mathcal{H}_{\text{hyper}}$. Since $T(z)$ is affine in z , the map $z \mapsto W(h_\eta(z)) = T(z) - I_n = (T_0 - I_n) + \sum_j z_j T_j$ is affine in z and therefore realizable by a ReLU network with linear output layer. The result follows from Theorem A.2.

Non-membership in $\mathcal{H}_{\text{concat}}$. Fix any non-empty open set $V \subset Z$. On $U \times V$, using (8) and (9):

$$f(s, z) = w^\top \left(T_0 + \sum_{j=1}^{d_z} z_j T_j \right) (As + a) + b.$$

This is a polynomial (in fact, at most degree 2) in (s, z) , hence smooth on $U \times V$. Its mixed second derivative is

$$\frac{\partial^2 f}{\partial s_i \partial z_j}(s, z) = w^\top T_j A e_i,$$

which is a *constant* that is nonzero by assumption (11).

By contrast, any function in $\mathcal{H}_{\text{concat}}$ is realized by a finite ReLU MLP on input $[s; z]$, which is continuous piecewise-linear (CPWL). A CPWL function is affine on each cell of a polyhedral partition, so its Hessian vanishes almost everywhere with respect to Lebesgue measure. Since $U \times V$ is open and non-empty, it has positive Lebesgue measure. But $\frac{\partial^2 f}{\partial s_i \partial z_j} \neq 0$ *everywhere* on $U \times V$, contradicting the almost-everywhere vanishing of the CPWL Hessian. Therefore $f \notin \mathcal{H}_{\text{concat}}$. \square

Remark A.4 (SAC with a shared hypernetwork-conditioned bottleneck adapter). Our implementation uses Soft Actor-Critic (SAC) [Haarnoja et al., 2018]. The actor produces a Gaussian policy by outputting $(\mu, \log \sigma)$; the critic outputs scalar action-values. A single bottleneck adapter is inserted after the trunk and before the output head in each network (Figure 6). In our implementation (Table 4), the trunk feature dimension is $d = 256$ and the bottleneck dimension is $k = 32$. The adapter includes a ReLU nonlinearity and has the form:

$$g_{\text{adapter}}(x; \omega) = W_{\text{up}}^{(\omega)} \cdot \text{ReLU}(W_{\text{down}}^{(\omega)} \cdot x), \quad (12)$$

where $W_{\text{down}}^{(\omega)} : \mathbb{R}^d \rightarrow \mathbb{R}^k$ and $W_{\text{up}}^{(\omega)} : \mathbb{R}^k \rightarrow \mathbb{R}^d$ are generated by a hypernetwork $h_\eta(z_t)$. The adapted features are $\tilde{x} = x + g_{\text{adapter}}(x; \omega)$ (skip connection; (3)). Following Beukman et al. [2023], we omit the activation function at the end of the trunk in both the actor and critic, so the adapter receives linear features. We share the same hypernetwork parameters η across the actor, critic, and dynamics adapters. Consequently, gradients from the dynamics objective update the shared mapping $z_t \mapsto \omega$, and the actor/critic can exploit this same context-to-parameter mapping through the shared adapters.

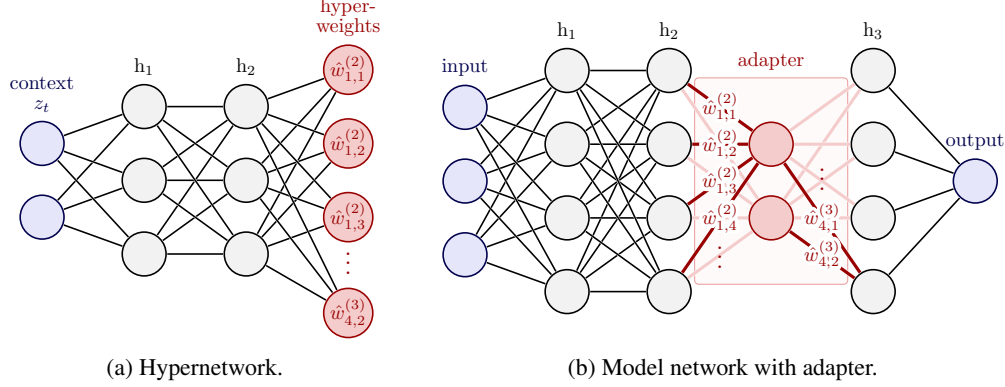


Figure 6: **Bottleneck adapter**. A hypernetwork (a) predicts parameters that are used within the dynamic model and RL networks (b).

Separation result for the bottleneck adapter architecture. The proof of Theorem A.1 uses a simplified linear adapter $g_{\text{adapter}}(x; \omega) = W^{(\omega)}x$ to isolate the core mechanism: context-dependent weights multiplying features to produce bilinear interactions in (x, z) . DMA*-SH uses the bottleneck adapter in (12), which instantiates the same weight-modulation principle and adds capacity through the intermediate nonlinearity. Define the bottleneck-adapter hypothesis class:

$$\mathcal{H}_{\text{hyper}}^{\text{bn}} : f(s, z) = w^\top (x(s) + g_{\text{adapter}}(x(s); h_\eta(z))) + b$$

with g_{adapter} as in (12). The linear adapter with full $W(\omega) \in \mathbb{R}^{d \times d}$ can represent adapter maps of rank up to d . The bottleneck adapter (12), even ignoring the ReLU nonlinearity, computes $W_{\text{up}}W_{\text{down}}x$ which has rank at most $k < d$. Therefore, an arbitrary full-rank linear adapter $W(\omega)x$ with $\text{rank}(W) > k$ cannot be represented by the bottleneck adapter. This means $\mathcal{H}_{\text{hyper}}^{\text{lin}} \not\subseteq \mathcal{H}_{\text{hyper}}^{\text{bn}}$ in general. Despite this, the specific witness function $f^*(s, z) = sz$ from Theorem A.1 is realizable by the bottleneck adapter, because the construction requires only rank 2.

Theorem A.5 (Separation for the bottleneck adapter). *Let $\mathcal{H}_{\text{hyper}}^{\text{bn}}$ be the bottleneck-adapter hypothesis class with bottleneck dimension $k \geq 2$, trunk dimension d , and trunk/hypernetwork realized by finite ReLU networks (with linear output for the hypernetwork). Let $\mathcal{H}_{\text{concat}}$ be as in Theorem A.1. Then*

$$\mathcal{H}_{\text{hyper}}^{\text{bn}} \not\subseteq \mathcal{H}_{\text{concat}}.$$

Proof. We show that the witness $f^*(s, z) = s \cdot z$ (with $d_s = d_z = 1$) belongs to $\mathcal{H}_{\text{hyper}}^{\text{bn}} \setminus \mathcal{H}_{\text{concat}}$.

Non-membership in $\mathcal{H}_{\text{concat}}$. This is proved in Theorem A.1 via the mixed-partial argument: $\frac{\partial^2 f^*}{\partial s \partial z} = 1 \neq 0$ everywhere, which is incompatible with CPWL functions on any domain with non-empty interior.

Membership in $\mathcal{H}_{\text{hyper}}^{\text{bn}}$. We exhibit an explicit construction with $k \geq 2$. Define the trunk as a ReLU network $x : \mathbb{R} \rightarrow \mathbb{R}^d$:

$$x(s) = (\text{ReLU}(s), \text{ReLU}(-s), 0, \dots, 0)^\top \in \mathbb{R}^d.$$

Note that $x_1(s) = \text{ReLU}(s) \geq 0$ and $x_2(s) = \text{ReLU}(-s) \geq 0$ for all $s \in \mathbb{R}$.

Define the hypernetwork output $\omega = h_\eta(z)$ to parameterize:

$$W_{\text{down}}^{(\omega)} : \mathbb{R}^d \rightarrow \mathbb{R}^k, \quad (W_{\text{down}})_{1,1} = 1, (W_{\text{down}})_{2,2} = 1, \text{ all other entries } 0. \quad (13)$$

$$W_{\text{up}}^{(\omega)} : \mathbb{R}^k \rightarrow \mathbb{R}^d, \quad (W_{\text{up}})_{1,1} = \alpha_1(z), (W_{\text{up}})_{2,2} = \alpha_2(z), \text{ all other entries } 0, \quad (14)$$

where $\alpha_1(z)$ and $\alpha_2(z)$ are to be determined.

Since $x_1(s), x_2(s) \geq 0$, and W_{down} selects these two coordinates, the input to the ReLU in (12) is

$$W_{\text{down}}x(s) = (x_1(s), x_2(s), 0, \dots, 0)^\top \in \mathbb{R}^k.$$

Since all components are non-negative, $\text{ReLU}(W_{\text{down}}x(s)) = W_{\text{down}}x(s)$. Thus the ReLU acts as the identity on these inputs, and the adapter computes:

$$g_{\text{adapter}}(x; \omega) = W_{\text{up}} W_{\text{down}} x(s) = (\alpha_1(z) x_1(s), \alpha_2(z) x_2(s), 0, \dots, 0)^\top.$$

With the skip connection, the adapted features are:

$$\tilde{x}(s) = x(s) + g_{\text{adapter}}(x(s); \omega) = ((1 + \alpha_1(z)) \text{ReLU}(s), (1 + \alpha_2(z)) \text{ReLU}(-s), 0, \dots, 0)^\top.$$

Setting $w = (1, 1, 0, \dots, 0)^\top$ and $b = 0$:

$$f(s, z) = w^\top \tilde{x}(s) = (1 + \alpha_1(z)) \text{ReLU}(s) + (1 + \alpha_2(z)) \text{ReLU}(-s).$$

Using $\text{ReLU}(s) - \text{ReLU}(-s) = s$, we require:

$$1 + \alpha_1(z) = z, \quad 1 + \alpha_2(z) = -z.$$

This gives $\alpha_1(z) = z - 1$ and $\alpha_2(z) = -(z + 1)$, both affine in z . The map $z \mapsto (\alpha_1(z), \alpha_2(z)) = (z - 1, -(z + 1))$ is affine and therefore realizable by a ReLU network with linear output. All other entries of W_{down} are context-independent constants (0 or 1), also trivially realizable.

Therefore $f(s, z) = z \cdot \text{ReLU}(s) - z \cdot \text{ReLU}(-s) = sz = f^*(s, z)$, and $f^* \in \mathcal{H}_{\text{hyper}}^{\text{bn}}$.

Combining Steps 1 and 2: $f^* \in \mathcal{H}_{\text{hyper}}^{\text{bn}} \setminus \mathcal{H}_{\text{concat}}$, hence $\mathcal{H}_{\text{hyper}}^{\text{bn}} \not\subseteq \mathcal{H}_{\text{concat}}$. \square

Theorem A.5 shows that the separation $\mathcal{H}_{\text{hyper}}^{\text{bn}} \not\subseteq \mathcal{H}_{\text{concat}}$ holds. The proof works precisely because the witness requires only rank 2, and the bottleneck ReLU acts as the identity on the relevant non-negative inputs. In general, the bottleneck ReLU can implement additional nonlinear transformations that the linear adapter cannot, so $\mathcal{H}_{\text{hyper}}^{\text{bn}}$ can represent functions beyond those in $\mathcal{H}_{\text{hyper}}^{\text{lin}}$ (provided the rank- k bottleneck suffices for the relevant subspace). While the blanket containment $\mathcal{H}_{\text{hyper}}^{\text{lin}} \subseteq \mathcal{H}_{\text{hyper}}^{\text{bn}}$ fails, the separation from $\mathcal{H}_{\text{concat}}$ transfers because the specific witness construction fits within the bottleneck’s capacity.

Table 9 shows that DMA*-SH outperforms the standard concatenation baseline Concat by 11.5% on eval-out, consistent with the architectural advantage identified by our results.

The separation results support a plausible mechanism for improved generalization. Hypernetworks can represent multiplicative interactions (including bilinear terms) exactly in simple cases, while concatenation MLPs typically approximate them via increasingly fine CPWL partitions. This approximation error may compound over trajectories, particularly in non-overlapping contexts where optimal policies differ drastically across modes.

Remark A.6 (Hypernetwork advantage for actuator inversion). Actuator inversion provides a concrete illustration of the expressive gap (Definition A.8). Suppose the environment contains a latent binary context $c \in \{-1, +1\}$, and the optimal policy satisfies $\pi^*(s, c) = c \cdot \pi_{\text{base}}(s)$, a sign flip in the action space. Since the agent does not observe c directly, DMA*-SH infers a continuous embedding $z_t = g_\phi(\tau_t^c)$ whose values cluster into two well-separated regions corresponding to the two actuator modes.

A hypernetwork can map these two embedding regions to adapter parameters $\omega^{(+)}$ and $\omega^{(-)}$ such that the induced mean actions satisfy $\mu_{\omega^{(-)}}(s) \approx -\mu_{\omega^{(+)}}(s)$. The adapter can induce opposite-signed mean actions by changing the effective mapping from trunk features to action means between the two modes (through the adapter-head composition), i.e., producing adapted features \tilde{x} such that $A\tilde{x}^{(-)} \approx -A\tilde{x}^{(+)}$ for the linear head matrix A . More generally, whenever the hypernetwork generates weights for a linear adapter layer, and these weights depend locally affinely on z , the resulting map is locally bilinear in (x, z) . By Theorem A.1, such bilinear structure cannot be represented exactly by concatenation ReLU MLPs.

Concatenation-based policies *can* approximate mode-switching behavior via CPWL partitioning: the network learns decision boundaries separating context regions and implements different linear maps on each piece. However, this indirect representation is more sensitive to encoder noise (small perturbations in z near a decision boundary cause large policy changes) and less parameter-efficient than direct multiplicative modulation. Table 2 confirms that DMA*-SH substantially outperforms concatenation baselines on non-overlapping environments where this architectural advantage is most critical.

Remark A.7 (Parameter, compute, and memory overhead). DMA*-SH introduces additional parameters through the hypernetwork, but the generated modulation is deliberately low-rank and shared. For trunk feature dimension d and adapter bottleneck dimension k , applying the adapter adds $O(dk)$ compute per adapted module, rather than the $O(d^2)$ cost of a full context-conditioned layer.

The most relevant practical comparison is to DA, the closest context-aware hypernetwork baseline. Measured under the same hardware, batch size, precision, and update schedule, DMA*-SH uses only 4.1% more parameters than DA, trains 43.5% faster, and uses 24.4% less peak memory. This indicates that sharing one context-to-adapter map across dynamics, actor, and critic is more efficient than maintaining separate hypernetwork pathways. Relative to simpler baselines without a hypernetwork, the cost is larger: compared with DMA, DMA*-SH uses 9.5% more parameters, trains 79.2% slower, and uses 21.7% more peak memory. We view this as the computational price of the dynamics-grounded operator modulation that drives the gains on non-overlapping contexts.

This overhead should be interpreted together with the expressivity result: concatenation-based ReLU networks can only approximate multiplicative context-dependent transformations through CPWL partitioning, whereas hypernetwork-conditioned adapters implement the relevant operator modulation directly. Empirically, DMA*-SH converges in comparable or fewer training steps than concatenation baselines (Figure 19) while achieving superior zero-shot performance on non-overlapping contexts (Table 8), indicating that the expressive advantage outweighs the parameter overhead.

A.2 Policy Overlap: Overlapping and Non-Overlapping Contexts

We categorize contextual families by the degree of *policy overlap*: whether a single context-unaware policy can achieve near-optimal performance across all contexts. We formalize this using a normalized worst-case regret criterion:

Definition A.8 (Overlapping and Non-Overlapping Contexts). Let \mathcal{C} be a set of contexts for environment E . For each $c \in \mathcal{C}$, let P^c be the transition dynamics and let π_c^* be an optimal policy achieving return $J_c(\pi) = \mathbb{E}_{\pi, P^c} [\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$. We normalize returns as $\bar{J}_c(\pi) := (J_c(\pi) - J_E^{\text{lo}}) / (J_E^{\text{hi}} - J_E^{\text{lo}})$, where J_E^{lo} and J_E^{hi} are fixed return bounds for E with $J_E^{\text{hi}} > J_E^{\text{lo}}$ (Table 5). We say \mathcal{C} is *non-overlapping* if there exists $\epsilon > 0$ such that for every context-unaware policy π ,

$$\max_{c \in \mathcal{C}} (\bar{J}_c(\pi_c^*) - \bar{J}_c(\pi)) \geq \epsilon.$$

Otherwise, \mathcal{C} is *overlapping*.

A policy is *context-unaware* if it conditions only on the observed state (e.g., $\pi(a | s)$) and does not take the context (or any inferred context representation) as input. Intuitively, in *overlapping* families there exist context-unaware policies whose worst-case normalized regret can be made arbitrarily small, i.e., a single robust policy can approach per-context optimal performance across the family. In *non-overlapping* families, optimal behaviors are mutually incompatible: every single context-unaware policy incurs a regret bounded away from zero in at least one context, and explicit context conditioning is necessary to be near-optimal across modes. Definition A.8 is conceptual; our overlapping and non-overlapping labels in Table 1 are operational, based on empirical behavior of strong context-unaware baselines and actuator-inversion incompatibility.

Remark A.9 (Non-overlapping functional modes in practice). Reversed wiring, swapped control channels, coordinate-frame mismatches, discrete gear-ratio shifts, and sim-to-real actuator remappings are natural examples of operator-switched dynamics that can exhibit non-overlap in the sense of Definition A.8. See also Section E.2.

We use actuator inversion as the canonical generator of non-overlapping structure (in the sense of Definition A.8) as it induces incompatible optimal policies across modes. Formally:

Definition A.10 (Actuator inversion). Assume the action space is sign-symmetric so that $c \cdot a \in \mathcal{A}$ for all $a \in \mathcal{A}$ and $c \in \{\pm 1\}$. A binary context $c \in \{\pm 1\}$ defines actuator-inverted dynamics by

$$P^c(s_{t+1} | s_t, a_t) = P(s_{t+1} | s_t, c \cdot a_t),$$

where P denotes the nominal dynamics. We focus on tasks where this inversion induces qualitatively different optimal control laws across $c = \pm 1$, for example when the optimal mean action satisfies a sign-flip relation across contexts: $\pi^*(s, -1) \approx -\pi^*(s, +1)$.

In the task class considered above, the optimal policies across modes often differ approximately by a sign flip. Hypernetwork-conditioned adapters can represent such mode-dependent transformations directly, whereas concatenation-based methods must approximate them via CPWL partitioning (Theorem A.1), a representation that is less robust when the inferred context z_t is noisy.

We examine how overlapping and non-overlapping context structures influence task difficulty and the stability of learned policies.

A.2.1 Context-Aware (e.g., Concat/DA Baselines)

- **Overlapping contexts (Easy)**

The policy class $\Pi_{\text{aware}} = \{\pi(a | s, c)\}$ is the set of all *context-aware* policies that depend on the ground-truth context c . Since the functions $\pi^*(s, c)$ are similar for different c , the agent can smoothly vary its behavior based on c . The complexity is effectively that of $|\mathcal{C}_{\text{train}}|$ separate policies, but shared structure across contexts can facilitate learning and enable generalization to $\mathcal{C}_{\text{eval-out}}$ via continuity.

- **Non-overlapping contexts (Solvable with context)**

When the agent has access to the ground-truth binary context $c \in \{\pm 1\}$, it can directly gate between mode-specific behaviors. In actuator inversion, the optimal policies across modes often satisfy a sign relation $\pi^*(s, -1) \approx -\pi^*(s, +1)$, which can be represented either by explicit gating on c or by mapping c to distinct adapter parameters. Hypernetwork-conditioned adapters provide a parameter-efficient way to implement mode-dependent transformations, but the key difficulty in our setting arises when the context must be inferred noisily (Section A.2.3).

A.2.2 Context-Unaware (e.g., Standard DR with Markovian Policy)

- **Overlapping contexts (Solvable)**

The policy class Π_{unaware} is the set of all *context-unaware* Markovian policies $\pi(a | s)$ that do not receive the context c , an inferred context embedding z_t , or trajectory history τ_t as input. If the contexts are overlapping, the optimal policies π_c^* for different c are similar. Therefore, a single policy $\pi \in \Pi_{\text{unaware}}$ can exist that is near-optimal for all $c \in \mathcal{C}$. The agent is effectively solving a single, slightly broader MDP.

- **Non-overlapping contexts (Inherently limited)**

By Definition A.8, every context-unaware Markovian policy $\pi(a | s)$ must incur normalized regret at least ϵ in some context $c \in \mathcal{C}$. Thus a state-only policy is forced to compromise between incompatible context-specific control laws.

Lemma A.11 (Failure of Markovian DR under actuator inversion). *Assume a symmetric action space and contexts $c \in \{\pm 1\}$ with dynamics $P^c(s' | s, a) = P(s' | s, c \cdot a)$. For a stochastic policy $\pi(a | s)$, define the negated policy $-\pi$ by $(-\pi)(a | s) \doteq \pi(-a | s)$. Assume the return satisfies the actuator-inversion symmetry $J_{-1}(\pi) = J_{+1}(-\pi)$ for all state-only Markovian policies π . Suppose further that there exists $\Delta \geq 0$ such that, for every $\pi \in \Pi_{\text{unaware}}$,*

$$J_{+1}(\pi) + J_{+1}(-\pi) \leq 2\Delta. \quad (15)$$

Then every every $\pi \in \Pi_{\text{unaware}}$ has domain-randomized objective

$$\mathbb{E}_{c \sim \text{Unif}\{\pm 1\}}[J_c(\pi)] \leq \Delta.$$

Consequently, if Δ is small compared with the per-context optima, then Markovian DR cannot achieve high average return.

Condition (15) is a sufficient condition for failure of Markovian DR. It formalizes the intuition that a policy and its negation cannot both make progress when the same action has opposite effects across contexts. This condition holds when reversing effective actuation reverses task progress, as in goal-reaching under actuator inversion. The empirical failure of DR on non-overlapping AIB environments (Table 2) is consistent with this analysis. Lemma 15 does not rule out recurrent or transformer DR policies that infer the hidden context from trajectory history; those are addressed empirically by the Amago baselines; see Appendix A.6 and Figure 15.

Proof. For any fixed $\pi \in \Pi_{\text{unaware}}$,

$$J_{\text{DR}}(\pi) = \frac{1}{2}(J_{+1}(\pi) + J_{-1}(\pi)) = \frac{1}{2}(J_{+1}(\pi) + J_{+1}(-\pi)) \leq \frac{1}{2} \cdot 2\Delta = \Delta,$$

where the second equality uses the actuator-inversion symmetry and the inequality uses condition (15). Since the bound holds for every such π , it also holds after maximizing over Π_{unaware} . \square

Remark A.12 (Context-unaware policies are epistemic POMDP solvers). When the policy is context-unaware (as in Markovian DR), the problem becomes an *epistemic POMDP* [Ghosh et al., 2021]: the true state is (s_t, c) , but the agent only observes s_t and must implicitly maintain a belief over the hidden context c . Thus, unknown contexts induce partial observability even when the raw state is fully observed. For overlapping contexts (e.g., small changes in mass or friction), the dynamics P^c vary smoothly. Small belief errors lead to small prediction errors, so the induced belief-MDP remains easy to optimize. For non-overlapping contexts (e.g., actuator inversion $c = \pm 1$), the dynamics for different contexts are mutually incompatible. Even slight uncertainty over c yields drastically different predictions for s_{t+1} . The context-averaged one-step kernel

$$\bar{P}(s_{t+1} \mid s_t, a_t) = \mathbb{E}_{c \sim \mathcal{C}_{\text{train}}} [P^c(s_{t+1} \mid s_t, a_t)],$$

can be mixture-like (often multimodal or high-variance). A context-unaware policy is therefore forced to average over contradictory behaviors, often producing low return. Maintaining a high-confidence belief under such conditions requires a sharp separation in the agent’s internal representation, a representation that is both difficult to learn and highly sensitive to noise, leading to higher optimization variance and poorer generalization.

Providing the agent with an accurately inferred context signal $z_t = g_\phi(\tau_t^c)$ (as in DMA*-SH) sidesteps the epistemic POMDP problem: the policy can condition directly on the correct mode instead of hedging across incompatible ones. This explains the dramatic performance gap of DMA*-SH on non-overlapping benchmarks (Table 2), while the advantage often disappears on overlapping benchmarks where even context-unaware baselines can succeed.

A.2.3 Context-Inferred (Our Method, DMA*-SH)

- *Overlapping* contexts (**Moderately difficult**)

The *context-inferred* policy class $\Pi_{\text{inferred}} = \{\pi(a \mid s, z)\}$ depends on the inferred context z . The agent must solve two coupled problems:

1. *Context inference*: infer z from a window of past K transitions $\tau = \{(s_k, a_k, \delta s_{k+1})\}$ via the encoder g_ϕ , and
2. *Control*: learn the policy $\pi(s, z)$.

Inference difficulty scales inversely with context distinguishability. Since the dynamics differ only mildly across contexts, the inferred representation z may be noisy or weakly informative. However, the control problem is comparatively easier: small errors in z induce only small policy deviations, so errors degrade performance smoothly.

- *Non-overlapping* contexts (**Very difficult**)

In our benchmark family, this is the most brittle setting for joint inference and control. Non-overlapping contexts provide strong statistical signals for inference (high Informativeness, e.g., large $I(\tau; c)$), so in principle the encoder can recover c from few transitions. In practice, however, even tiny inference errors are catastrophic: misclassifying $c = +1$ as $c = -1$ induces the *opposite* control law, and the agent immediately fails. Learning is therefore brittle unless the encoder $g_\phi(\tau)$ produces accurate embeddings on most episodes. This creates a difficult credit-assignment loop during joint training.

Encoder imprecision may arise from finite window size K (partial observability), stochasticity in P^c (e.g., sensor noise), or approximation limits of g_ϕ . In non-overlapping regimes, such small errors are amplified severely in RL performance (e.g., through error propagation in value targets or large policy regret), since the failure modes are binary with no “graceful” degradation. The brittleness is worse for concatenation-based baselines, which must learn hard boundaries in their inputs, whereas hypernetwork-conditioning (as in DMA*-SH) effectively captures the multiplicative structure of actuator inversion (Theorem A.1).

Remark A.13 (The smoothness inductive bias of latent-variable models in meta-RL). In variational latent-variable meta-RL methods for POMDPs (e.g., VariBAD [Zintgraf et al., 2020]), the agent infers

a latent context z from experience and regularizes this inference with an explicit prior. Typically, the optimization involves an ELBO of the form

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(z|\tau)}[\log p_\theta(\tau | z)] - \text{KL}(q_\phi(z | \tau) \| p(z)),$$

where both the posterior $q_\phi(z | \tau)$ and the prior $p(z)$ are *unimodal* Gaussians. The KL term encourages $q_\phi(z | \tau)$ to stay close to the prior $p(z)$, which in common instantiations is a unimodal Gaussian (e.g., $p(z) = \mathcal{N}(0, I)$).

In actuator inversion, good representations benefit from mapping the two modes to well-separated latent regions (e.g., approximately opposite signs). A unimodal Gaussian prior places most mass near the origin, so KL regularization biases the encoder toward posteriors that allocate non-negligible probability to intermediate latent values that do not correspond cleanly to either mode. This mismatch can yield unstable training signals: small errors in z_t can induce the wrong action sign and large performance drops. This is consistent with the empirical failures of VariBAD on ODE and DI tasks (Figure 18).

In contrast, DMA*-SH avoids smooth latent priors by representing context through multiplicative hypernetwork modulation. Given an inferred embedding z_t , the hypernetwork generates adapter weights $\omega = h_\eta(z_t)$. This allows the policy and critic to implement sharp mode-dependent transformations, e.g., approximating a sign flip between the two mode clusters,

$$\pi_{\xi, \omega}(s) \approx -\pi_{\xi, \omega'}(s),$$

which matches the multiplicative structure of actuator inversion. These multiplicative interactions provide an inductive bias that is often helpful in non-overlapping regimes where ELBO-regularized latent models struggle empirically.

A.3 Informativeness, Variability, and Gradient Variance

We analyze the geometry of learned context representations using two complementary diagnostics: *Variability* (representation spread) and *Informativeness* (mutual information between representation and context). We then link Variability to learning stability by deriving a policy-gradient variance bound controlled by Variability. For binary–continuous contexts, this variance-based lens also motivates an *implicit structural information bottleneck* interpretation: preserving mode information while selectively compressing within-mode variation.

A.3.1 Decomposing Variability into Noise, Compression, and Separation

We relate the Variability of inferred embeddings z (Section 6.4) to the geometry of the learned representation space. In non-overlapping AIB tasks, the context can be factored as $C = (S, U)$ where $S \in \{\pm 1\}$ is the actuator-inversion mode and U is a continuous parameter (e.g., mass in DI). Below, we treat (S, U, \mathcal{T}, Z) as random variables under the rollout distribution induced by a fixed policy (i.e., at a fixed training iterate), with lowercase denoting realizations (e.g., τ is a realization of \mathcal{T} and z of Z). \mathcal{T} denotes a trajectory window of length K and $Z \doteq g_\phi(\mathcal{T}) \in \mathbb{R}^{d_z}$ is the inferred embedding.

Fix one dataset split $M \in \{\text{train, eval-in, eval-out}\}$, and let \mathcal{D}_M denote the induced empirical distribution over trajectory windows obtained by pooling all contexts in that split. Each window $\mathcal{T} \sim \mathcal{D}_M$ is generated by the policy interacting with the CMDP under ground-truth context (S, U) . All expectations below are with respect to this split-induced joint distribution. Variability is the average per-coordinate variance of embeddings pooled across the split:

Definition A.14 (Variability). The Variability of split M is

$$\text{Variability}(M) \doteq \frac{1}{d_z} \sum_{i=1}^{d_z} \text{Var}[Z_i] = \frac{1}{d_z} \text{tr}(\text{Cov}[Z]). \quad (16)$$

Theorem A.15 (Variability decomposition for binary-continuous contexts). *Let $C = (S, U)$ be the ground-truth context with $S \in \{\pm 1\}$ binary and U a continuous (possibly vector-valued) random variable. Let $Z = g_\phi(\mathcal{T}) \in \mathbb{R}^{d_z}$ be the embedding of a trajectory window $\mathcal{T} \sim \mathcal{D}_M$. Assume $\mathbb{E}\|Z\|_2^2 < \infty$. Define $\bar{z}(s, u) \doteq \mathbb{E}[Z | S = s, U = u]$, $\mu_s \doteq \mathbb{E}_{U|S=s}[\bar{z}(s, U)]$, and $p \doteq \mathbb{P}(S = +1)$.*

Then

$$\begin{aligned}
\text{Variability}(M) &= \underbrace{\frac{1}{d_z} \mathbb{E}_{S,U} \left[\text{tr} \left(\text{Cov}(Z \mid S, U) \right) \right]}_{\text{Term 1: within-context noise}} \\
&\quad + \underbrace{\frac{1}{d_z} \text{tr} \left(\mathbb{E}_S \left[\text{Cov}_{U|S}(\bar{z}(S, U)) \right] \right)}_{\text{Term 2: within-mode variation (continuous)}} \\
&\quad + \underbrace{\frac{p(1-p)}{d_z} \|\mu_+ - \mu_-\|_2^2}_{\text{Term 3: between-mode separation (binary)}}. \tag{17}
\end{aligned}$$

Moreover, if $u \mapsto \bar{z}(s, u)$ is L_s -Lipschitz on $\text{supp}(U \mid S = s)$, then

$$\frac{1}{d_z} \text{tr} \left(\text{Cov}_{U|S=s}(\bar{z}(s, U)) \right) \leq \frac{L_s^2}{d_z} \mathbb{E}[\|U - \mathbb{E}[U \mid S = s]\|_2^2 \mid S = s]. \tag{18}$$

If, additionally, $\bar{z}(s, \cdot)$ is differentiable on a convex set $\mathcal{U}_s \supseteq \text{supp}(U \mid S = s)$, then $L_s \leq \sup_u \|\partial \bar{z}(s, u) / \partial u\|_{\text{op}}$.

Remark A.16 (Geometric Intuition). The decomposition (17) is a statistical identity relating total embedding variance to three interpretable components. Its validity is independent of the encoder g_ϕ or the learning algorithm, and holds for any random variable Z with the (S, U, Z) joint structure. Term 1 represents the *representation noise floor*, Term 2 quantifies the *within-mode variation* induced by U (and hence the extent to which a representation does *not* collapse within-mode differences), and Term 3 measures the *separation* between binary actuator modes. Later, in Theorem A.17, we formalize how $\text{Variability}(M)$ (and hence its decomposition) enters an upper bound on policy-gradient variance.

Our proofs rely on the law of total covariance. For any random vector X and conditioning random variable Y :

$$\text{Cov}(X) = \mathbb{E}_Y [\text{Cov}(X \mid Y)] + \text{Cov}_Y(\mathbb{E}[X \mid Y]). \tag{19}$$

Furthermore, we utilize the following independent-copy identity for the trace of the covariance: If $X \in \mathbb{R}^d$ is a random vector with finite second moments ($\mathbb{E}\|X\|_2^2 < \infty$), and X' is an independent copy of X , then

$$\text{tr}(\text{Cov}(X)) = \mathbb{E}\|X - \mathbb{E}X\|_2^2 = \frac{1}{2} \mathbb{E}\|X - X'\|_2^2. \tag{20}$$

To verify this, note that since X' is independent of X and identically distributed:

$$\begin{aligned}
\mathbb{E}\|X - X'\|_2^2 &= \mathbb{E}\|X\|_2^2 - 2\mathbb{E}[X]^\top \mathbb{E}[X'] + \mathbb{E}\|X'\|_2^2 \\
&= 2\mathbb{E}\|X\|_2^2 - 2\|\mathbb{E}[X]\|_2^2 \\
&= 2 \text{tr}(\text{Cov}(X)).
\end{aligned}$$

$\|\cdot\|_{\text{op}}$ denotes the induced L_2 operator norm, and $\text{supp}(P)$ denotes the support of the distribution P .

Proof. All expectations and covariances are under the distribution $\mathcal{T} \sim \mathcal{D}_M$ and induced (S, U) .

Apply (19) to Z conditioned on (S, U) . With $\bar{z}(S, U) = \mathbb{E}[Z \mid S, U]$,

$$\text{Cov}(Z) = \mathbb{E}_{S,U} [\text{Cov}(Z \mid S, U)] + \text{Cov}_{S,U}(\bar{z}(S, U)).$$

Taking $\frac{1}{d_z} \text{tr}(\cdot)$ and using $\text{tr}(\mathbb{E}[A]) = \mathbb{E}[\text{tr}(A)]$,

$$\text{Variability}(M) = \frac{1}{d_z} \mathbb{E}_{S,U} \left[\text{tr} \left(\text{Cov}(Z \mid S, U) \right) \right] + \frac{1}{d_z} \text{tr} \left(\text{Cov}_{S,U}(\bar{z}(S, U)) \right). \tag{21}$$

The first term in (21) is Term 1 in (17).

Apply (19) to $\bar{z}(S, U)$ conditioned on S :

$$\text{Cov}_{S,U}(\bar{z}(S, U)) = \mathbb{E}_S [\text{Cov}_{U|S}(\bar{z}(S, U))] + \text{Cov}_S(\mu_S),$$

where $\mu_S = \mathbb{E}_{U|S}[\bar{z}(S, U)]$. Note that for each s , $\mu_s = \mathbb{E}_{U|S=s}[\bar{z}(s, U)] = \mathbb{E}[Z \mid S = s]$ by iterated expectation. Taking $\frac{1}{d_z} \text{tr}(\cdot)$ gives Term 2 in (17) plus the binary term.

For $S \in \{+1, -1\}$ with $\mathbb{P}(S = +1) = p$, let $\mu_{\pm} = \mu_{S=\pm 1}$ and $m = p\mu_+ + (1-p)\mu_-$. Then $\mu_+ - m = (1-p)(\mu_+ - \mu_-)$ and $\mu_- - m = -p(\mu_+ - \mu_-)$, so

$$\begin{aligned} \text{Cov}_S(\mu_S) &= p(\mu_+ - m)(\mu_+ - m)^\top + (1-p)(\mu_- - m)(\mu_- - m)^\top \\ &= [p(1-p)^2 + (1-p)p^2](\mu_+ - \mu_-)(\mu_+ - \mu_-)^\top \\ &= p(1-p)(\mu_+ - \mu_-)(\mu_+ - \mu_-)^\top. \end{aligned}$$

Hence $\frac{1}{d_z} \text{tr}(\text{Cov}_S(\mu_S)) = \frac{p(1-p)}{d_z} \|\mu_+ - \mu_-\|_2^2$, which is Term 3 in (17).

Lipschitz bound. Fix s and let $f(u) = \bar{z}(s, u)$. By assumption, f is L_s -Lipschitz on $\text{supp}(U | S = s)$. Thus for all u, v , $\|f(u) - f(v)\|_2 \leq L_s \|u - v\|_2$. Let U' be an independent copy of U under $U | S = s$. Applying the independent-copy identity (20) and the Lipschitz property, we obtain:

$$\begin{aligned} \text{tr}(\text{Cov}_{U|S=s}(f(U))) &= \frac{1}{2} \mathbb{E}[\|f(U) - f(U')\|_2^2 | S = s] \\ &\leq \frac{L_s^2}{2} \mathbb{E}[\|U - U'\|_2^2 | S = s] = L_s^2 \mathbb{E}[\|U - \mathbb{E}[U | S = s]\|_2^2 | S = s]. \end{aligned}$$

Dividing by d_z , we obtain (18). When $\bar{z}(s, \cdot)$ is differentiable on a convex $\mathcal{U}_s \supseteq \text{supp}(U | S = s)$, the mean value inequality implies $L_s \leq \sup_{u \in \mathcal{U}_s} \|\partial \bar{z}(s, u) / \partial u\|_{\text{op}}$. \square

Within-mode compression and between-mode separation. Theorem A.15 decomposes $\text{Variability}(M)$ into three nonnegative contributions with distinct geometric meanings:

- **Term 1: within-context noise (window-level spread).** The first term $\frac{1}{d_z} \mathbb{E}_{S,U}[\text{tr}(\text{Cov}(Z | S, U))]$ measures the average dispersion of embeddings across different trajectory windows at *fixed* ground-truth context (S, U) . It captures stochasticity arising from partial observability, rollout randomness, finite window length, and encoder noise. So this term is the *noise floor* that blurs any geometric organization induced by $\bar{z}(S, U)$. Operationally, reducing Term 1 can improve consistency of the context signal across trajectory windows.
- **Term 2: within-mode mean-variation (compression of U).** The second term $\frac{1}{d_z} \text{tr}(\mathbb{E}_S[\text{Cov}_{U|S}(\bar{z}(S, U))])$ quantifies how the mean embeddings $\bar{z}(S, U)$ move as the continuous parameter U varies *within* each actuator mode S . By (18), small Lipschitz sensitivity L_s of $u \mapsto \bar{z}(s, u)$ guarantees that the within-mode mean-variation is controlled by L_s^2 times the intrinsic spread of $U | S = s$. Intuitively, when L_s is small, varying u within a fixed mode s can shift the mean embedding $\bar{z}(s, u)$ only marginally, so the mean representation is nearly “flat” in the U -direction. In this sense, *compressing* U amounts to making the within-mode mean embedding insensitive to changes in u . Separately, Proposition A.19 shows that perfect information-theoretic compression $I(Z; U | S) = 0$, is sufficient but not necessary for Term 2 = 0.
- **Term 3: between-mode mean-separation (separation across S).** The third term $\frac{p(1-p)}{d_z} \|\mu_+ - \mu_-\|_2^2$ (where $\mu_s = \mathbb{E}_{U|S=s}[\bar{z}(s, U)]$) is the binary mode *separation* induced by actuator inversion. Operationally, larger Term 3 signifies stronger mode separation, increasing as the actuator modes occupy more distinct regions in the representation space. Terms 2 and 3 can move in opposite directions: the encoder can simultaneously compress U within each mode (small Term 2) while separating the actuator modes (large Term 3).

A.3.2 Variability controls policy-gradient variance

We establish a formal connection between the Variability decomposition (Theorem A.15) and policy-gradient variance, and use this connection to resolve the apparent Informativeness–Variability paradox in Section 6.4 (Figure 3).

Theorem A.17 (Policy-gradient variance bound controlled by Variability). *Let the ground-truth context be $C = (S, U)$ with $S \in \{\pm 1\}$ and U a continuous random variable. Let $Z = g_\phi(\mathcal{T}) \in \mathbb{R}^{d_z}$ be the inferred embedding and let $G \in \mathbb{R}^{d_z}$ denote a single-sample policy-gradient estimator. Assume:*

- (A1) *The embedding Z and gradient G have finite second moments: $\mathbb{E}\|Z\|_2^2 < \infty$ and $\mathbb{E}\|G\|_2^2 < \infty$.*

(A2) *The conditional gradient variance is bounded: there exists $V_0 \geq 0$ such that $\mathbb{E}[\text{tr}(\text{Cov}(G | Z))] \leq V_0$.*

(A3) *The conditional mean gradient $m(z) \doteq \mathbb{E}[G | Z = z]$ is L_m -Lipschitz on a compact set \mathcal{Z} containing $\text{supp}(Z)$.*

Then the total gradient variance satisfies:

$$\text{tr}(\text{Cov}(G)) \leq V_0 + L_m^2 \cdot d_z \cdot \text{Variability}(M). \quad (22)$$

Under Assumptions (A1)–(A3), Theorem A.17 upper-bounds the variance of the single-sample policy-gradient estimator by an irreducible term V_0 (capturing conditional sampling noise given Z) plus a term proportional to $\text{Variability}(M)$. These are regularity conditions under the split-induced distribution \mathcal{D}_M : (A1) is plausible since SimNorm constrains embeddings to a compact set [Obando-Ceron et al., 2025] and SAC uses bounded actions with gradient clipping [Haarnoja et al., 2018]; (A2) captures conditional stochasticity of single-sample gradients; (A3) assumes bounded sensitivity of $m(z)$ to changes in z , a standard regularization perspective [Gouk et al., 2021]. Accordingly, we treat V_0 and L_m as problem-dependent constants. We use Theorem A.17 as an interpretive upper bound linking representation spread to policy-gradient variance, not as a tight characterization of SAC’s optimization dynamics. Figure 8 reports proxy diagnostics consistent with lower gradient-norm variability under DMA*-SH.

Proof. All expectations and covariances are taken under the split-induced distribution $\mathcal{T} \sim \mathcal{D}_M$ and the induced random variables (S, U, Z, G) (conditional on fixed current parameters).

By Assumption (A1), $\text{Cov}(G)$ is well-defined. Similarly, $\text{Cov}(m(Z))$ is well-defined because $\|m(Z)\|_2^2$ is integrable; specifically, by Jensen’s inequality and Assumption (A1),

$$\mathbb{E}\|m(Z)\|_2^2 = \mathbb{E}\|\mathbb{E}[G | Z]\|_2^2 \leq \mathbb{E}\mathbb{E}\|G\|_2^2 | Z] = \mathbb{E}\|G\|_2^2 < \infty.$$

Applying the law of total covariance to G conditioned on Z :

$$\begin{aligned} \text{Cov}(G) &= \underbrace{\mathbb{E}[\text{Cov}(G | Z)]}_{\text{irreducible sampling noise even if } Z \text{ is fixed}} + \underbrace{\text{Cov}(\mathbb{E}[G | Z])}_{\text{variance coming from } Z \text{ moving around}} \\ &= \mathbb{E}[\text{Cov}(G | Z)] + \text{Cov}(m(Z)). \end{aligned} \quad (23)$$

Taking traces and using linearity: $\text{tr}(\text{Cov}(G)) = \mathbb{E}[\text{tr}(\text{Cov}(G | Z))] + \text{tr}(\text{Cov}(m(Z)))$.

By Assumption (A2):

$$\text{tr}(\text{Cov}(G)) \leq V_0 + \text{tr}(\text{Cov}(m(Z))). \quad (24)$$

Let Z' be an independent copy of Z . Applying the independent-copy identity (20) with $X = m(Z)$:

$$\text{tr}(\text{Cov}(m(Z))) = \frac{1}{2} \mathbb{E}[\|m(Z) - m(Z')\|_2^2].$$

By the Lipschitz Assumption (A3): $\|m(Z) - m(Z')\|_2^2 \leq L_m^2 \|Z - Z'\|_2^2$, hence $\text{tr}(\text{Cov}(m(Z))) \leq \frac{L_m^2}{2} \mathbb{E}[\|Z - Z'\|_2^2]$.

Since $\mathbb{E}\|Z\|_2^2 < \infty$ by Assumption (A1), we may apply (20) with $X = Z$:

$$\text{tr}(\text{Cov}(m(Z))) \leq L_m^2 \cdot \text{tr}(\text{Cov}(Z)). \quad (25)$$

Substituting into (24):

$$\text{tr}(\text{Cov}(G)) \leq V_0 + L_m^2 \cdot \text{tr}(\text{Cov}(Z)). \quad (26)$$

By Definition A.14, $\text{tr}(\text{Cov}(Z)) = d_z \cdot \text{Variability}(M)$. Substituting into (26) yields the stated bound (22). \square

A.3.3 The Informativeness–Variability paradox: When less information improves learning

Informativeness $I(Z; S, U)$ measures how much information Z carries about the *full* context $C = (S, U)$. In non-overlapping AIB tasks, only part of that context may be *decision-relevant* for stable control and learning:

1. The binary mode S is decision-critical (wrong sign can catastrophically flip the control law).
2. The continuous coordinate U (e.g., mass) may be weakly relevant or even largely irrelevant *within each mode* over the benchmark range, especially if robust control can handle that variation.

Intuitively, one might expect that embeddings capturing more information about the full context (S, U) would yield better RL performance. Proposition A.18 shows this intuition can fail: increasing informativeness can increase gradient variance by encoding within-mode variation that is not needed for control. This creates a paradoxical regime where an embedding with strictly *lower* informativeness can achieve a strictly *tighter* policy-gradient variance bound and can therefore learn more stably. Proposition A.18 formalizes this possibility:

Proposition A.18 (The Informativeness–Variability paradox). *Let the setting be as in Theorems A.15 and A.17. Assume:*

(B1) *The binary mode is recoverable from the embedding: $H(S | Z) = 0$.*

(B2) *The embedding carries nontrivial information about the continuous parameter beyond the mode: $I(Z; U | S) > 0$.*

Define the mode-collapsed embedding $\tilde{Z} \in \mathbb{R}^{d_z}$ by $\tilde{Z} \doteq \mu_S$, where μ_s is as defined in Theorem A.15. Then:

1. Informativeness strictly *decreases*: $I(\tilde{Z}; S, U) < I(Z; S, U)$.
2. Variability strictly *decreases* by removing the within-mode term: with $\tilde{z}(s, u) \doteq \mathbb{E}[\tilde{Z} | S = s, U = u] = \mu_s$,

$$\text{tr}(\text{Cov}(\tilde{Z})) = p(1-p)\|\mu_+ - \mu_-\|_2^2 \leq \text{tr}(\text{Cov}(Z)),$$

and the inequality is strict whenever Term 1 or Term 2 in (17) is positive.

3. The policy-gradient variance upper bound improves despite lower Informativeness: *If the policy-gradient estimator computed with \tilde{Z} satisfies the same assumptions of Theorem A.17 with the same constants V_0 and L_m , then*

$$\text{tr}(\text{Cov}(G_{\tilde{Z}})) \leq V_0 + L_m^2 \text{tr}(\text{Cov}(\tilde{Z})) < V_0 + L_m^2 \text{tr}(\text{Cov}(Z)),$$

whenever the inequality in Part 2 is strict. Thus, a representation can be less informative about the full context (S, U) while yielding a strictly smaller policy-gradient variance upper bound.

Proposition A.18 demonstrates that the Informativeness–Variability paradox is mathematically possible: an embedding with strictly lower Informativeness can yield a strictly tighter gradient-variance bound. This provides one explanation consistent with our empirical observations in Section 6.4 (Figure 3), where DMA*-SH achieves lower Informativeness than DMA* and DMA yet superior RL performance. The proposition constructs an idealized oracle embedding $\tilde{Z} = \mu_S$ that perfectly collapses within-mode variation. We do not claim DMA*-SH learns this oracle exactly; rather, architectural constraints (SimNorm, shared hypernetwork, dynamics-only training) create pressure toward similar compression (Appendix A.3.4). The empirical MI trajectories in Figure 7 are consistent with this interpretation and with Assumptions (B1)–(B2): $I(Z; S)$ saturates early near $H(S)$ (high mode recoverability, equivalently small $H(S | Z)$), while $I(Z; U | S)$ remains nonzero but relatively reduced for DMA*-SH (partial within-mode compression).

Proof. Part 1: Informativeness decreases. Since $\tilde{Z} = \mu_S$ is a deterministic function of S , $I(\tilde{Z}; U | S) = 0$. By the chain rule for mutual information, $I(\tilde{Z}; S, U) = I(\tilde{Z}; S) + I(\tilde{Z}; U | S) = I(\tilde{Z}; S) \leq H(S)$. By Assumption (B1), $H(S | Z) = 0$, hence $I(Z; S) = H(S) - H(S | Z) = H(S)$ and

$I(Z; S, U) = I(Z; S) + I(Z; U | S) = H(S) + I(Z; U | S)$. By Assumption (B2), $I(Z; U | S) > 0$. So $I(\tilde{Z}; S, U) \leq H(S) < H(S) + I(Z; U | S) = I(Z; S, U)$.

Part 2: Variability decreases. By construction, $\tilde{Z} = \mu_S$ depends only on S , hence for all (s, u) ,

$$\text{Cov}(\tilde{Z} | S = s, U = u) = 0, \quad \tilde{z}(s, u) \doteq \mathbb{E}[\tilde{Z} | S = s, U = u] = \mu_s.$$

Since $\tilde{z}(s, u) = \mu_s$ is constant in u , $\text{Cov}_{U|S=s}(\tilde{z}(s, U)) = 0$. Applying the decomposition (Theorem A.15) to \tilde{Z} :

$$\text{tr}(\text{Cov}(\tilde{Z})) = 0 + 0 + p(1-p)\|\mu_+ - \mu_-\|_2^2 = p(1-p)\|\mu_+ - \mu_-\|_2^2.$$

Applying the same decomposition to Z :

$$\begin{aligned} \text{tr}(\text{Cov}(Z)) &= \mathbb{E}_{S,U} \left[\text{tr}(\text{Cov}(Z | S, U)) \right] + \text{tr} \left(\mathbb{E}_S [\text{Cov}_{U|S}(\bar{z}(S, U))] \right) + p(1-p)\|\mu_+ - \mu_-\|_2^2 \\ &= d_z \cdot \text{Term 1} + d_z \cdot \text{Term 2} + p(1-p)\|\mu_+ - \mu_-\|_2^2. \end{aligned}$$

Since Term 1 and Term 2 are non-negative, $\text{tr}(\text{Cov}(\tilde{Z})) \leq \text{tr}(\text{Cov}(Z))$, with strict inequality whenever Term 1 > 0 or Term 2 > 0 .

Part 3: Variance bound improves although Informativeness drops. By Theorem A.17, the upper bound for $G_{\tilde{Z}}$ is $V_0 + L_m^2 \text{tr}(\text{Cov}(\tilde{Z}))$, and the upper bound for G_Z is $V_0 + L_m^2 \text{tr}(\text{Cov}(Z))$. By Part 2, $\text{tr}(\text{Cov}(\tilde{Z})) < \text{tr}(\text{Cov}(Z))$ whenever Term 1 or Term 2 is positive. Therefore:

$$V_0 + L_m^2 \text{tr}(\text{Cov}(\tilde{Z})) < V_0 + L_m^2 \text{tr}(\text{Cov}(Z)),$$

proving that the policy-gradient variance upper bound strictly improves for \tilde{Z} despite its lower informativeness. \square

A.3.4 DMA*-SH as an Implicit Structural Information Bottleneck

We formalize the notion of a *structural information bottleneck* (SIB) for binary–continuous contexts, and explain how DMA*-SH can be viewed as *implicitly* implementing an *approximate SIB* via dynamics prediction under architectural constraints. We emphasize that SIB serves here as an interpretive framework, not an optimization target: DMA*-SH does *not* explicitly optimize an SIB objective. Rather, the SIB perspective helps explain the emergent representation geometry analyzed in Appendix A.3.5: In particular, the observed signatures (fast mode acquisition and reduced within-mode spread in certain tasks) are consistent with a capacity-constrained representation that prioritizes mode information S over within-mode detail U when this is sufficient for dynamics prediction and control.

Two complementary notions of compression. We first formalize what we mean by “compression”.

(A) Variance: compression of within-mode mean dependence on U . Recall the conditional mean embedding $\bar{z}(s, u) \doteq \mathbb{E}[Z | S = s, U = u]$ and mode-conditional mean $\mu_s \doteq \mathbb{E}_{U|S=s}[\bar{z}(s, U)]$. Term 2 in Theorem A.15 measures how much the conditional mean embedding $\bar{z}(s, u)$ varies with U within each mode s . Small Term 2 indicates that $\bar{z}(s, u)$ is approximately “flat” along U -directions within a fixed mode, a variance-based signature of compression. Moreover, Theorem A.15 provides a quantitative sufficient condition: if $u \mapsto \bar{z}(s, u)$ is L_s -Lipschitz, then

$$\text{Term 2} \leq \frac{1}{d_z} \sum_{s \in \{\pm 1\}} \mathbb{P}(S = s) \cdot L_s^2 \cdot \text{tr}(\text{Cov}(U | S = s)). \quad (27)$$

Thus, small Lipschitz constants imply small within-mode mean variation.

(B) Information: compression of U given S . A complementary notion is the conditional mutual information $I(Z; U | S)$. Smaller $I(Z; U | S)$ means that, after conditioning on the mode S , the representation Z retains less information about the continuous parameter U .

Relationship between the two notions. Proposition A.19 below establishes that these two notions are related but not equivalent: perfect information-theoretic compression ($I(Z; U | S) = 0$) implies zero within-mode mean variation (Term 2 = 0), but the converse fails because Term 2 captures only *mean* dependence on U , not higher-moment dependence.

Proposition A.19 (Boundary behavior of within-mode variation). *Let the setting be as in Theorem A.15.*

1. If $I(Z; U | S) = 0$, then Term 2 = 0.
2. The converse does not hold: Term 2 = 0 does not imply $I(Z; U | S) = 0$.

Proof. Part 1. By definition, $I(Z; U | S) = 0$ if and only if $Z \perp U | S$, meaning $p(Z | S = s, U = u) = p(Z | S = s)$ for all s, u . This implies $\bar{z}(s, u) = \mathbb{E}[Z | S = s, U = u] = \mathbb{E}[Z | S = s] = \mu_s$ for all u . Since $\bar{z}(s, u) = \mu_s$ is constant in u , we have $\text{Cov}_{U|S=s}(\bar{z}(s, U)) = 0$ for each s . Averaging over S and taking $\frac{1}{d_z} \text{tr}(\cdot)$ yields Term 2 = 0.

Part 2. We construct a counterexample. Let $\bar{z}(s, u) = \mu_s$ be constant in u , but let the conditional variance depend on u . Suppose $U \sim \text{Uniform}[1, 2]$ and $Z | (S = s, U = u) \sim \mathcal{N}(\mu_s, u^2 I_{d_z})$. Then $\bar{z}(s, u) = \mu_s$ is constant in u , so $\text{Cov}_{U|S}(\bar{z}(s, U)) = 0$ and Term 2 = 0. However, since $p(Z | S = s, U = u)$ depends on u through the variance, we have $Z \not\perp U | S$, hence $I(Z; U | S) > 0$. \square

Thus Term 2 is best interpreted as a *mean-flatness* signature rather than a complete proxy for $I(Z; U | S)$.

Structural information bottleneck (SIB). The classical information bottleneck (IB) seeks representations that preserve task-relevant information while compressing other variation [Tishby et al., 2000, Alemi et al., 2017, Banerjee and Montúfar, 2020]. In actuator-inversion regimes, the context $C = (S, U)$ has an asymmetric structure: flipping the mode S typically changes the *qualitative* controller structure (e.g., action sign), while U modulates *within-mode* quantitative details (e.g., gains). This motivates a conditional compression objective that prioritizes mode information while penalizing within-mode sensitivity to U .

Definition A.20 (Structural information bottleneck (SIB) for actuator inversion). Let $C = (S, U)$ with $S \in \{\pm 1\}$ binary and U continuous and $Z = g_\phi(\mathcal{T})$. The *SIB objective* is

$$\mathcal{L}_{\text{SIB}}(Z) = I(Z; S) - \beta I(Z; U | S), \quad \beta > 0, \quad (28)$$

or in constrained form: maximize $I(Z; S)$ over g_ϕ subject to $I(Z; U | S) \leq R$. We say that Z implements an *exact SIB* if $I(Z; S) = H(S)$ and $I(Z; U | S) = 0$, and an *approximate SIB* if $I(Z; S)/H(S)$ (assuming $H(S) > 0$, as in our non-degenerate splits) is high and $I(Z; U | S)$ is relatively low.

The first term $I(Z; S)$ measures *mode sufficiency*: how much information the representation retains about the binary mode. The second term $I(Z; U | S)$ measures *within-mode compression*: how much information about U is retained beyond mode identification. The SIB objective prioritizes mode information while penalizing within-mode encoding.

Implicit SIB via dynamics prediction and architectural constraints. DMA*-SH does not optimize (28) explicitly. The encoder g_ϕ is trained solely via the dynamics prediction loss (2),

$$L_{\phi, \theta, \eta} = \|\delta \hat{s}_{t+1} - \delta s_{t+1}\|_2^2,$$

where δs_{t+1} is the next-state change and $\delta \hat{s}_{t+1}$ is its prediction. This objective contains no information-theoretic regularizer. Nevertheless, the combination of dynamics prediction with architectural constraints creates an *implicit* bottleneck that can approximate the SIB structure. The architectural constraints in DMA*-SH are shown in Table 3.

These constraints limit the effective capacity of the representation. Without additional assumptions, minimizing squared prediction error is *not* generally equivalent to maximizing mutual information. We use the following only as an *interpretive* lens: Under restricted representation classes, optimizing dynamics prediction tends to favor encoding those context factors whose recovery most reduces prediction error. In actuator-inversion environments, the mode S typically governs the sign/compatibility

Table 3: Architectural constraint and effect on Z .

Constraint	Effect
SimNorm	Constrains Z to product of $L = 2$ simplices of dimension $V = 4$
Low d_z	Limited embedding dimensionality ($d_z = 8$)
LSTM bottleneck	Sequential processing compresses temporal information
Random masking	Forces robustness, prevents overfitting to specific features

of the action-to-dynamics response, whereas U often modulates within-mode scaling. Under strong capacity constraints, it is therefore plausible that optimization pressure prioritizes (or learns earlier) the mode information yielding high $I(Z; S)$ while compressing fine-grained within-mode dependence on U , yielding relatively low $I(Z; U | S)$. The key insight is that DMA*-SH’s compression is not imposed by the loss, but by the architecture: the dynamics loss determines *what* information is useful; the architectural constraints determine *how much* can be retained. When mode information is more predictive for dynamics than within-mode variation, the resulting representation can be biased toward high $I(Z; S)$ and reduced $I(Z; U | S)$, thereby approximating the SIB objective in effect.

SIB links representation geometry to policy-gradient variance. The SIB framework connects to our main theoretical results as follows. Theorem A.15 decomposes total Variability(M) into Term 1 (within-context noise), Term 2 (within-mode mean-variation of $\bar{z}(s, u) = \mathbb{E}[Z | S = s, U = u]$ across $U | S$), and Term 3 (between-mode mean-separation).

By Proposition A.19, $I(Z; U | S) = 0$ implies Term 2 = 0; more generally, reduced conditional informativeness about U is compatible with (though does not in itself imply) smaller within-mode mean-variation. By Theorem A.17, under Lipschitz regularity of $m(z) = \mathbb{E}[G | Z = z]$, the policy-gradient variance satisfies $\text{tr}(\text{Cov}(G)) \leq V_0 + L_m^2 d_z \cdot \text{Variability}(M)$; since this bound depends on Variability(M), reducing Term 2 while preserving Term 3 tightens this bound, provided Term 1 does not increase to offset the reduction. Proposition A.18 formalizes the resulting tradeoff: an embedding with lower total informativeness $I(Z; S, U)$ can achieve a tighter gradient-variance upper bound when the reduction comes from compressing within-mode structure rather than degrading mode information.

In this sense, DMA*-SH can be viewed as approximately solving a *structural IB* that prioritizes mode information while compressing within-mode variation in a way that benefits the policy-gradient variance bound.

Why DMA*-SH outperforms Concat on non-overlapping eval-out. The SIB framework highlights three properties. To state them precisely, we distinguish between the data-generating process and what the agent observes. The latent context $C = (S, U)$ determines the MDP dynamics under which rollouts are generated. For fixed parameters, this induces the causal structure $(S, U) \rightarrow \mathcal{T} \rightarrow Z \rightarrow \omega$, where $Z = g_\phi(\mathcal{T})$ and $\omega = h_\eta(Z)$. Let X denote additional non-context inputs to downstream modules (e.g., state for the policy or state–action pairs for the critic). Downstream modules receive ω together with X without ever observing (S, U) .

- (i) *Mode sufficiency*: $H(S | Z) \approx 0$, i.e., the latent mode S is approximately recoverable from Z .
- (ii) *Within-mode compression*: $I(Z; U | S)$ is relatively small, i.e., Z carries limited information about U beyond what S provides.
- (iii) *Downstream context-channel restriction (architectural)*: any downstream output \hat{Y} produced by a module that receives context only through (Z, ω) satisfies

$$\hat{Y} \perp (S, U) | (Z, X).$$

Consequently, any sensitivity to within-mode shifts in U must be mediated through how U affects the observed rollout distribution and the inferred embedding.

Condition (iii) is an architectural property of DMA*-SH: the shared adapter pathway ensures that downstream modules receive context only through $\omega = h_\eta(Z)$, with no explicit (S, U) input. Conditions (i) and (ii) are learned properties; gradient detachment of ω in the RL losses ensures that the encoder is shaped only by dynamics prediction, creating pressure to encode features predictive for dynamics (primarily mode S) while architectural bottlenecks limit capacity for within-mode detail.

Downstream outputs can still depend on U marginally through the observed non-context inputs X (e.g., the state), since the rollout distribution under $P^{(S,U)}$ depends on U . We therefore interpret (ii)–(iii) as a structural bias against *explicit* and potentially brittle U -conditioning.

In non-overlapping tasks, the binary mode S acts as a compatibility variable: correct behavior must commit to the appropriate mode, while in our benchmark ranges U predominantly modulates within-mode quantitative details. The standard context-aware baseline Concat receives (S, U) directly as input to the policy, so it has no architectural restriction analogous to (iii) and no intrinsic pressure to *ignore* U . In practice, this can entangle policy/value features with fine-grained U -dependence learned on $\mathcal{C}_{\text{train}}$, which may be brittle under the held-out split $\mathcal{C}_{\text{eval-out}}$. Concat is thus context-aware but *not context-selective*. DMA*-SH’s architectural restriction (iii), combined with approximate satisfaction of (i)–(ii), provides a structural bias toward mode-based rather than fine-grained U -based conditioning, which we interpret as contributing to its improved robustness on non-overlapping $\mathcal{C}_{\text{eval-out}}$ (Table 8).

Non-overlapping versus overlapping environments. The SIB interpretation applies most naturally to non-overlapping (actuator-inversion) environments where the binary mode S has a discontinuous, dominant effect on dynamics. In overlapping environments with purely continuous context $C = (U_1, U_2)$ and no binary mode, there is no Term 3 (between-mode separation) since no discrete mode exists. In such environments, compression can discard task-relevant variation, and the actuator-inversion-specific inductive bias provides less benefit. This is consistent with DMA*-SH achieving competitive but not dominant performance on overlapping environments (Table 9).

A.3.5 Regime shifts and selective-compression signatures on $\mathcal{C}_{\text{eval-out}}$

Figure 7 reports $I(Z; S)$, $I(Z; U | S)$, Variability, and returns on the held-out split $\mathcal{C}_{\text{eval-out}}$ for three non-overlapping environments: DI, Cartpole, and Reacher (Hard). Mutual information (MI) is measured in nats; for a binary mode $S \in \{\pm 1\}$ with balanced splits, the maximum is $I(Z; S) = H(S) = \ln 2 \approx 0.693$ nats. We treat MI estimates as comparative diagnostics (relative ordering across methods under the same estimator and sample budget), not calibrated absolute values.

Across all three environments, the evolution exhibits a consistent qualitative ordering:

1. **Rapid mode acquisition.** $I(Z; S)$ increases quickly and enters a near-saturated regime (relative to $H(S)$) early in optimization, indicating rapid acquisition of mode information. In Cartpole and Reacher, $I(Z; S)$ approaches values close to $\ln 2$, whereas in DI the attained level is lower but still substantial.
2. **Slower within-mode refinement.** $I(Z; U | S)$ increases more gradually and typically continues to rise after $I(Z; S)$ has effectively plateaued. The eventual level of $I(Z; U | S)$ is task-dependent, with Reacher exhibiting markedly larger within-mode information than DI or Cartpole, consistent with arm-length scaling being more consequential for Reacher’s kinematics/dynamics than mass (DI) or pole length (Cartpole) for the other two tasks.
3. **Regime changes aligned with mode saturation.** Visible kinks (regime changes) occur around the time $I(Z; S)$ enters its near-saturated regime. The accompanying behavior of Variability(M) and returns is environment-specific.

DI. DI exhibits the sharpest association between mode acquisition and task success. Returns rise sharply and reach their maximum precisely when $I(Z; S)$ transitions from rapid growth to a near-plateau. This temporal alignment is consistent with mode identification being a key enabler of success in DI: under actuator inversion, incorrect mode identification induces acceleration in the wrong direction, which (under sparse reward) yields near-zero return. After this mode-acquisition transition, returns remain saturated, while $I(Z; U | S)$ and Variability continue to evolve on $\mathcal{C}_{\text{eval-out}}$, indicating that additional within-mode information is not required to improve returns in this environment. While our results do not claim a monotone relationship between returns and mutual information, this pattern is compatible with the tradeoff highlighted by Proposition A.18: increasing within-mode information can, in principle, increase representational variability (and hence loosen the gradient-variance upper bound) without yielding additional control benefit once mode information is reliable.

Cartpole. In Cartpole, $I(Z; S)$ reaches a near-saturated regime early, while returns continue improving over a longer horizon. The Variability curve exhibits a regime change aligned with the

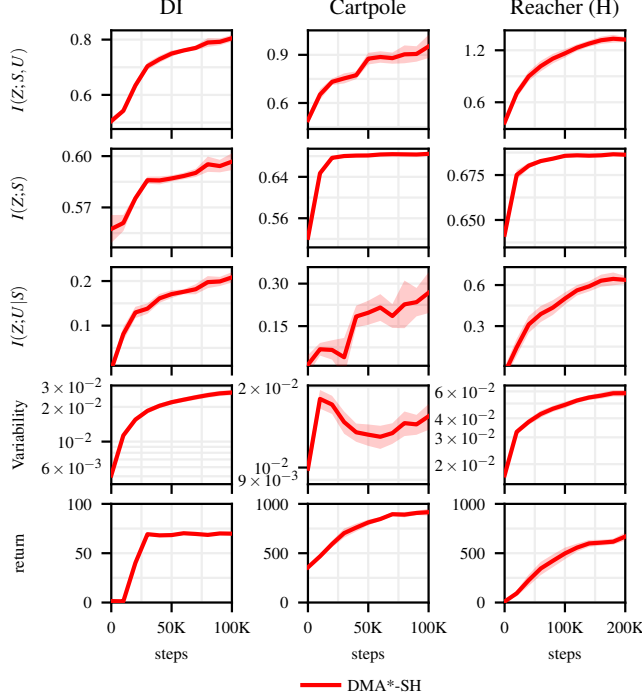


Figure 7: **Held-out dynamics of mode sufficiency vs. within-mode refinement.** Evolution of $I(Z;S,U)$, $I(Z;S)$, $I(Z;U|S)$, Variability, and returns for DI, Cartpole, and Reacher (Hard). Across tasks, mode information $I(Z;S)$ rises fast and approaches the binary ceiling $H(S) = \ln 2$, while within-mode information $I(Z;U|S)$ accumulates more slowly; kinks near mode saturation coincide with regime changes in Variability, consistent with a *mode-first, task-adaptive structural bottleneck*.

mode-acquisition transition, followed by non-monotone evolution thereafter. This is consistent with a two-stage process: (i) early acquisition of mode information sufficient to avoid catastrophic inversion errors, followed by (ii) longer-horizon refinement associated with within-mode adaptation to U (pole length) under feedback control. Since $I(Z;S)$ does not decrease (and in fact increases slightly) across this transition, a post-transition decrease in Variability cannot be attributed to a loss of mode information; rather, it indicates that non-mode contributions to Variability must be decreasing over that interval (i.e., a reduction in Term 1 and/or Term 2 in Theorem A.15). Moreover, the observation that $I(Z;U|S)$ can increase while Variability decreases is compatible with Proposition A.19 (Part 2): within-mode information about U can be expressed through aspects of the conditional distribution beyond the conditional mean map $\bar{z}(s, u)$, so Term 2 need not track $I(Z;U|S)$. Unlike DI, returns in Cartpole rise gradually rather than jumping at the mode transition, reflecting that balancing provides continuous reinforcement even when mode discrimination remains imperfect, and feedback control can partially offset suboptimal gains.

Reacher (Hard). In Reacher (Hard), $I(Z;S)$ again enters a near-saturated regime early, while returns continue to improve substantially afterwards, in parallel with a sustained rise of $I(Z;U|S)$. This pattern is consistent with a regime where mode identification resolves the actuator-inversion incompatibility, yet accurate within-mode control remains sensitive to arm-length scaling (U), which affects the reachable workspace, the joint-to-end-effector geometry, and the torques required to move limbs of varying length and inertia. Correspondingly, within-mode information appears more behaviorally consequential than in DI. The Variability curve exhibits a regime change aligned with the kink in $I(Z;S)$ and continues to increase before stabilizing late in optimization, consistent with extended refinement on held-out contexts.

Taken together, the eval-out trajectories are qualitatively consistent with a *mode-first* optimization pattern and an *approximate, task-adaptive* structural bottleneck: $I(Z;S)$ approaches $H(S)$ early (mode sufficiency), while $I(Z;U|S)$ continues to grow later (within-mode refinement) in a task-

dependent manner. When U is strongly control-relevant, DMA*-SH does *not* compress it aggressively; instead it learns it. The alignment between mode-acquisition transitions and regime changes in Variability is consistent with the variance decomposition of Theorem A.15 and the representation-dependent component of the policy-gradient variance bound in Theorem A.17. Finally, the differing coupling between $I(Z; U | S)$ and returns across environments is compatible with the tradeoff articulated by Proposition A.18. We emphasize that this SIB interpretation is a post-hoc explanatory framework supported by these empirical patterns.

A.4 Scale Control and Directional Geometry in the Context Space

We record two empirically relevant properties of DMA* and DMA*-SH that motivate cosine-based geometric summaries used later (Definition A.21) and help interpret the DI visualizations (Appendix A.4.2) through the decomposition in Theorem A.15 (Appendix A.3).

The DMA* normalization pipeline controls representation scale. DMA* and DMA*-SH apply (i) per-sample input normalization via AvgL1Norm and (ii) output normalization via SimNorm (Section 4). AvgL1Norm rescales each input vector $x \in \mathbb{R}^N$ to $\text{AvgL1Norm}(x) = \frac{Nx}{\sum_i |x_i|}$, which removes arbitrary per-sample magnitude drift in the encoder input. SimNorm maps z_t to a product of simplices via group-wise softmax, yielding a bounded output space with controlled scale. Together, these operators substantially reduce uncontrolled radial scaling of embeddings compared to an unnormalized encoder, so that angular geometry (cosine similarity) becomes a more interpretable notion of similarity for per-context mean embeddings.

Shared hypernetwork conditioning is associated with stronger directional concentration in DI. DMA*-SH conditions a single hypernetwork h_η on the normalized embedding z_t to generate adapter weights used across dynamics, policy, and value networks. In DI, the cosine heatmaps and RO scores (Figures 4 and 3) show that moving from DMA* to DMA*-SH is associated with higher within-mode cosine similarity across mass values, while the actuator-inversion modes remain distinguishable in both cosine structure and t-SNE. This empirical pattern is consistent with Theorem A.15: reduced continuous within-mode spread (compression) can co-occur with nontrivial separation across the binary actuator mode.

A.4.1 Representation-Overlap (RO).

We introduce Representation-Overlap (RO) as a directional summary of how per-context mean embeddings align in latent space. This is particularly natural in DMA*-SH as SimNorm controls the scale of z_t and the hypernetwork h_η is conditioned on the normalized embedding (Appendix A.4), so cosine-based comparisons are less confounded by uncontrolled radial scaling. Accordingly, cosine similarity provides an invariant notion of similarity under positive rescalings:

$$\cos(\alpha u, \beta v) = \cos(u, v) \quad \forall \alpha, \beta > 0.$$

When norms are controlled by normalization, higher cosine similarity between mean embeddings is a useful proxy for more directionally concentrated representations, and therefore for more similar hypernetwork conditioning inputs.

Definition A.21 (Representation-Overlap (RO)). Let $\{c^{(1)}, \dots, c^{(n)}\}$ be n distinct contexts. For each $c^{(i)}$, given B embeddings $\{z_{c^{(i)}}^{(b)}\}_{b=1}^B$, define the mean embedding $\mu_{c^{(i)}} = \frac{1}{B} \sum_{b=1}^B z_{c^{(i)}}^{(b)}$. The pairwise cosine similarity is

$$\cos(\mu_{c^{(i)}}, \mu_{c^{(j)}}) = \frac{\mu_{c^{(i)}}^\top \mu_{c^{(j)}}}{\|\mu_{c^{(i)}}\| \|\mu_{c^{(j)}}\|}. \quad (29)$$

The global RO score averages over all n^2 pairs (including $i = j$):

$$\text{RO} = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \cos(\mu_{c^{(i)}}, \mu_{c^{(j)}}). \quad (30)$$

RO summarizes global directional concentration of the per-context mean embeddings. In the DI setting, Theorem A.15 (Appendix A.3) distinguishes (i) within-mode spread over the continuous

parameter and (ii) separation across actuator-inversion modes. Empirically, higher RO in DMA*-SH co-occurs with stronger within-mode alignment across mass (a directional signature consistent with reduced continuous within-mode spread), while still preserving a clear distinction between actuator modes (Appendix A.4.2).

A.4.2 t-SNE Visualization and RO Cosine Similarity Analysis

We visualize the geometry of inferred embeddings z_t in DI using t-SNE [Van der Maaten and Hinton, 2008] and cosine similarity of per-context means (Definition A.21). These views complement the scalar diagnostics in Figure 3 by illustrating how the mean structure across contexts changes, which is central to the decomposition in Theorem A.15 (Appendix A.3).

Figure 4 shows t-SNE visualizations for DI comparing DMA, DMA*, and DMA*-SH. Each point corresponds to an embedding $z_t = g_\phi(\tau)$, and colors indicate the underlying context pair (actuator-inversion mode and mass). Across methods, embeddings organize into two actuator-mode groups, while DMA*-SH exhibits visibly weaker stratification by mass within each mode. This qualitative pattern is consistent with Theorem A.15: lowering the continuous within-mode contribution (compression of U within fixed S) can reduce overall Variability, while retaining a nontrivial between-mode separation term for actuator inversion.

We further examine cosine similarities between per-context mean embeddings using Equation 29 (Figure 4, bottom).

DMA \rightarrow DMA*: **Effect of normalization.** In DI, DMA exhibits strongly negative cosines between means from opposite actuator modes (approximately -0.4 to -0.97), indicating near sign-opposition in mean directions across modes. With input/output normalization (DMA*), these cross-mode cosines move to near-orthogonality (about 0.01 – 0.03), yielding a less extreme directional relationship between the two actuator-mode groups.

DMA* \rightarrow DMA*-SH: **Effect of shared hypernetwork conditioning.** Moving from DMA* to DMA*-SH, within-mode cosines across different mass values increase (roughly from 0.53 – 1.0 to values near 1.0 in DI), consistent with stronger within-mode directional concentration and hence reduced continuous within-mode spread (the compression term in Theorem A.15). At the same time, cross-mode cosines remain distinct from within-mode values (e.g., around 0.14 – 0.23 in DI), consistent with maintaining a separable actuator-inversion structure rather than collapsing the discrete distinction.

Overall, the t-SNE and cosine patterns provide a geometric complement to Figure 3 and are consistent with the compression/separation decomposition in Theorem A.15.

A.5 Implicit Regularization via Shared Hypernetwork Gradients

Shared hypernetworks (DMA*-SH). We examine the effect of sharing a single hypernetwork across the dynamics model, policy, and Q-function in DMA*-SH. The training loop in Algorithm 2 interleaves (i) RL updates of the actor and critic base parameters and (ii) dynamics-driven updates of the representation and hypernetwork parameters:

- RL updates:

$$\xi \leftarrow \xi - \alpha_1 \nabla_\xi \sum_c L_\xi^c, \quad \zeta \leftarrow \zeta - \alpha_2 \nabla_\zeta \sum_c L_\zeta^c,$$

where L_ξ^c and L_ζ^c are the actor and critic losses computed using $\pi_{\xi,\omega}$ and $Q_{\zeta,\omega}$ with $\omega = h_\eta(z_t)$ (treated as a constant in L_ξ^c and L_ζ^c via stop-gradient).

- Dynamics updates:

$$\phi \leftarrow \phi - \alpha_3 \nabla_\phi \sum_c L_{\phi,\theta,\eta}^c, \quad \theta \leftarrow \theta - \alpha_3 \nabla_\theta \sum_c L_{\phi,\theta,\eta}^c, \quad \eta \leftarrow \eta - \alpha_3 \nabla_\eta \sum_c L_{\phi,\theta,\eta}^c,$$

where $L_{\phi,\theta,\eta}^c$ is the forward-dynamics (FD) reconstruction objective (2) and depends on $f_{\theta,\omega}(s_t, a_t)$ with $\omega = h_\eta(z_t)$.

Since ω is detached in the RL losses, the RL objectives update only the base parameters ξ and ζ ; gradients from L_ξ and L_ζ do not backpropagate through ω to η (or to z_t) (Figure 1b). Nonetheless, the objectives remain coupled through the *shared adapter pathway*, namely the forward mapping

$$\tau_t^c \xrightarrow{g_\phi} z_t \xrightarrow{h_\eta} \omega_t \xrightarrow{\text{shared adapters}} \{f_{\theta, \omega_t}, \pi_{\xi, \omega_t}, Q_{\zeta, \omega_t}\}. \quad (31)$$

The same $\omega_t = h_\eta(z_t)$ is reused by the dynamics model, actor, and critic. During RL updates, ξ and ζ are optimized under the adapter configuration produced by the current hypernetwork, but gradients through ω_t are stopped. During dynamics updates, ϕ , θ , and η are refined by the reconstruction objective using data collected by the agent. This sharing injects a dynamics-trained structural constraint into RL optimization: the actor and critic can use context only through adapters shaped by the dynamics objective. Appendix A.6 contrasts this factorization with history-based recurrent and Transformer agents.

Separate hypernetworks (DMA*-H). We compare DMA*-SH to a separate-hypernetwork design, denoted DMA*-H, where the dynamics model, policy, and Q-function have separate hypernetwork parameters η^f, η^π, η^Q and produce separate adapter weights $\omega_t^f, \omega_t^\pi, \omega_t^Q$:

$$z_t \xrightarrow{h_\eta^m} \omega_t^m \xrightarrow{\text{adapter}} M_{\omega_t^m}^m, \quad m \in \{f, \pi, Q\}. \quad (32)$$

Here M^m denotes the corresponding module: f_θ, π_ξ , or Q_ζ . The FD loss updates only η^f , while the actor and critic losses update η^π and η^Q . This decouples dynamics-driven and reward-driven adaptation, which can introduce objective mismatch when reward-driven adapters move in directions not supported by the dynamics-trained pathway.

Shadow gradients (diagnostics in η -space and z -space). Since ω is detached in the RL losses in DMA*-SH, the true training-time gradients through the shared adapter pathway (31) satisfy $\nabla_\eta L_\pi = 0$ and $\nabla_z L_\pi = 0$ by construction. For analysis only, we define a *shadow* computation that temporarily removes the stop-gradient through ω and differentiates RL losses along this pathway. These shadow quantities are *not* used for training and do not affect the optimization trajectory; they are diagnostic probes of how the RL objective would prefer to change the shared adapter pathway if the detachment were removed.

In η -space, the shadow sensitivity of L_{RL} to the shared hypernetwork parameters follows $\frac{\partial L_{\text{RL}}}{\partial \eta} = \frac{\partial L_{\text{RL}}}{\partial \omega} \cdot \frac{\partial \omega}{\partial \eta}$, where $\frac{\partial \omega}{\partial \eta} = \frac{\partial h_\eta(z_t)}{\partial \eta}$. The factor $\frac{\partial L_{\text{RL}}}{\partial \omega}$ chains through both $\pi_{\xi, \omega}$ and $Q_{\zeta, \omega}$. For instance, for the actor, the relevant component is $-\frac{\partial}{\partial \omega} \mathbb{E}_{\alpha_t \sim \pi_{\xi, \omega}} [Q_{\zeta, \omega} - \alpha \log \pi_{\xi, \omega}]$. Although η is shielded from RL updates to avoid direct interference with the FD-trained adapter mapping, the shadow gradients quantify how the RL objective would (hypothetically) prefer to adjust the shared mapping $z \mapsto \omega$ if such updates were enabled, while training instead proceeds by adapting (ξ, ζ) under the dynamics-trained adapters.

In z -space, the analogous shadow sensitivity is $\frac{\partial L_{\text{RL}}}{\partial z_t} = \frac{\partial L_{\text{RL}}}{\partial \omega} \cdot \frac{\partial \omega}{\partial z_t}$, where $\frac{\partial \omega}{\partial z_t} = \frac{\partial h_\eta(z_t)}{\partial z_t}$. The factor $\frac{\partial L_{\text{RL}}}{\partial \omega}$ chains through both $\pi_{\xi, \omega}$ and $Q_{\zeta, \omega}$, so $\|\nabla_{z_t} L_\pi\|$ and $\|\nabla_{z_t} L_Q\|$ quantify how strongly the policy and critic objectives would (hypothetically) prefer to change the inferred context signal along the adapter pathway. These z -space norms provide a direct measure of *context sensitivity* in the shared embedding space \mathbb{R}^{d_z} .

Gradient analysis (Figure 8). We report gradient-based diagnostics across DI, DI-Friction, ODE, and Cartpole to characterize how DMA*-SH and DMA*-H differ in (i) effective context sensitivity along the adapter pathway and (ii) the variance of selected gradient-norm signals.

In DMA*-SH, all quantities involving $\nabla_\eta L_\pi$, $\nabla_z L_\pi$, or the cosines with these terms are computed using the shadow construction described above.

1. Mean policy hypernetwork sensitivity (η -space): $\mathbb{E}\|\nabla_\eta L_\pi\|$ (DMA*-SH) and $\mathbb{E}\|\nabla_{\eta^\pi} L_\pi\|$ (DMA*-H). Across environments, the DMA*-SH curve remains substantially larger than DMA*-H, indicating a persistent hypothetical tendency of the policy objective to reshape the shared hypernetwork mapping if such updates were enabled. This is consistent with the actor and critic adapting their base parameters (ξ, ζ) under a dynamics-trained adapter mapping rather than directly rewriting it.

2. Mean policy context sensitivity (z -space): $\mathbb{E}\|\nabla_z L_\pi\|$. DMA*-SH exhibits consistently larger values than DMA*-H, indicating stronger dependence of the policy objective on the inferred context through the adapter pathway. In DMA*-H, the smaller values are consistent with weaker effective context utilization along this route.
3. Variance of policy gradients w.r.t. context (z -space): $\text{Var}\|\nabla_z L_\pi\|$. Unlike the mean sensitivity, which captures persistent dependence on the adapter pathway, the relative ordering varies across environments, suggesting that the variance is influenced by environment-specific nonstationarity and mode-switching.
4. Variance of policy base-parameter gradient norms (ξ -space): $\text{Var}\|\nabla_\xi L_\pi\|$. DMA*-SH typically shows slightly lower variance than DMA*-H, consistent with a modestly more stable optimization signal for the policy base parameters under shared adapters.
5. Variance of context-encoder gradients under dynamics (ϕ -space): $\text{Var}\|\nabla_\phi L_d\|$. This measures variance of the dynamics-driven learning signal entering the context encoder. Values are small in magnitude, but DMA*-SH often shows slightly lower variance than DMA*-H.
6. Variance of dynamics gradients w.r.t. embedding (z -space): $\text{Var}\|\nabla_z L_d\|$. This measures variance of the dynamics objective’s sensitivity to the inferred embedding in \mathbb{R}^{d_z} . Magnitudes are small, so this metric serves mainly as a weak supporting diagnostic.
7. Dynamics–policy alignment (η -space): $\cos(\nabla_\eta L_d, \nabla_\eta L_\pi)$ (DMA*-SH) and a heuristic analogue in DMA*-H comparing η^f and η^π . In DMA*-SH, this cosine tracks whether the (hypothetical) reward-driven direction in the shared η coordinates tends to align with or oppose the dynamics-driven direction. In DMA*-H, the corresponding cosine is a heuristic as it compares different parameter spaces.
8. Policy–critic alignment (η -space): $\cos(\nabla_\eta L_\pi, \nabla_\eta L_Q)$ (DMA*-SH) and a heuristic analogue in DMA*-H comparing η^π and η^Q . In DMA*-SH, this cosine summarizes whether actor and critic objectives would push the shared hypernetwork parameters in similar directions under the hypothetical update. In DMA*-H, the corresponding cosine is a heuristic as it compares different parameter spaces.
9. Returns. DMA*-SH attains faster learning and higher returns than DMA*-H across the considered environments. This performance gap co-occurs with the separation in $\mathbb{E}\|\nabla_\eta L_\pi\|$ and $\mathbb{E}\|\nabla_z L_\pi\|$, consistent with stronger and more persistent context dependence through the shared adapter pathway.

Overall, the most consistent separation is in the mean context-sensitivity diagnostics, both $\mathbb{E}\|\nabla_\eta L_\pi\|$ and $\mathbb{E}\|\nabla_z L_\pi\|$. This suggests that DMA*-SH maintains a sustained context-dependent adapter pathway shaped by the dynamics-trained hypernetwork, whereas DMA*-H exhibits substantially weaker sensitivity along the corresponding policy pathway.

A.6 Comparison to History-Based Recurrent and Transformer Agents

A history-based agent such as Amago-2 [Grigsby et al., 2024b] uses a trajectory encoder or sequence model to produce a latent state shared by actor and critic heads. Abstractly, this adaptation pathway can be written as

$$\tau_t \xrightarrow{\text{SeqModel}_\psi} h_t \xrightarrow{\text{policy/value heads}} (a_t, Q_t).$$

Thus task inference and control are coupled inside a single hidden state h_t optimized by RL objectives. Such models can in principle implement multiplicative interactions through attention or gating, and our expressivity theorem is not a separation against Transformers. The distinction is structural rather than purely expressive.

In DMA*-SH, the corresponding pathway is the dynamics-grounded factorization in (31). History first determines a context embedding z_t , then a hypernetwork maps this embedding to a shared adapter configuration ω_t , and the same ω_t is reused by the dynamics model, actor, and critic. The map $z_t \mapsto \omega_t$ is trained only by the forward-dynamics objective, while actor and critic gradients are stopped through ω_t . Thus all context dependence seen by control is forced through one dynamics-trained intermediate object, rather than an arbitrary RL-trained hidden state.

This factorization separates two roles that are often conflated in recurrent or Transformer baselines: *mode identification* and *mode-conditioned computation*. A sequence model may infer latent task

information from history, but it does not require the inferred representation to parameterize the transition model and the control networks through the same operator. DMA*-SH imposes this constraint explicitly: the multiplicative operator used by control is grounded in transition prediction and shared across model, actor, and critic.

This distinction is particularly important in non-overlapping settings. A wrong mode estimate can make the policy catastrophically wrong, so early reward gradients may be sparse, high-variance, or misleading. Dynamics prediction provides a dense supervised signal from the first transition. Detaching ω_t forces the actor and critic to learn under a dynamics-trained adapter configuration rather than rewriting the context-to-operator map themselves. Hence detachment is part of the structural prior, not merely an implementation detail.

No-detach ablation. Our no-detach ablations in Figure 15 support this interpretation. Allowing critic gradients through the shared hypernetwork degrades performance, while allowing actor gradients through the shared hypernetwork causes near-zero returns on the tested non-overlapping environments. In contrast, analogous actor-gradient flow in DMA* without a shared hypernetwork has only modest effect. This indicates that the critical failure mode is not simply that RL gradients touch a context encoder, but that RL gradients rewrite the shared context-to-operator map used by all modules.

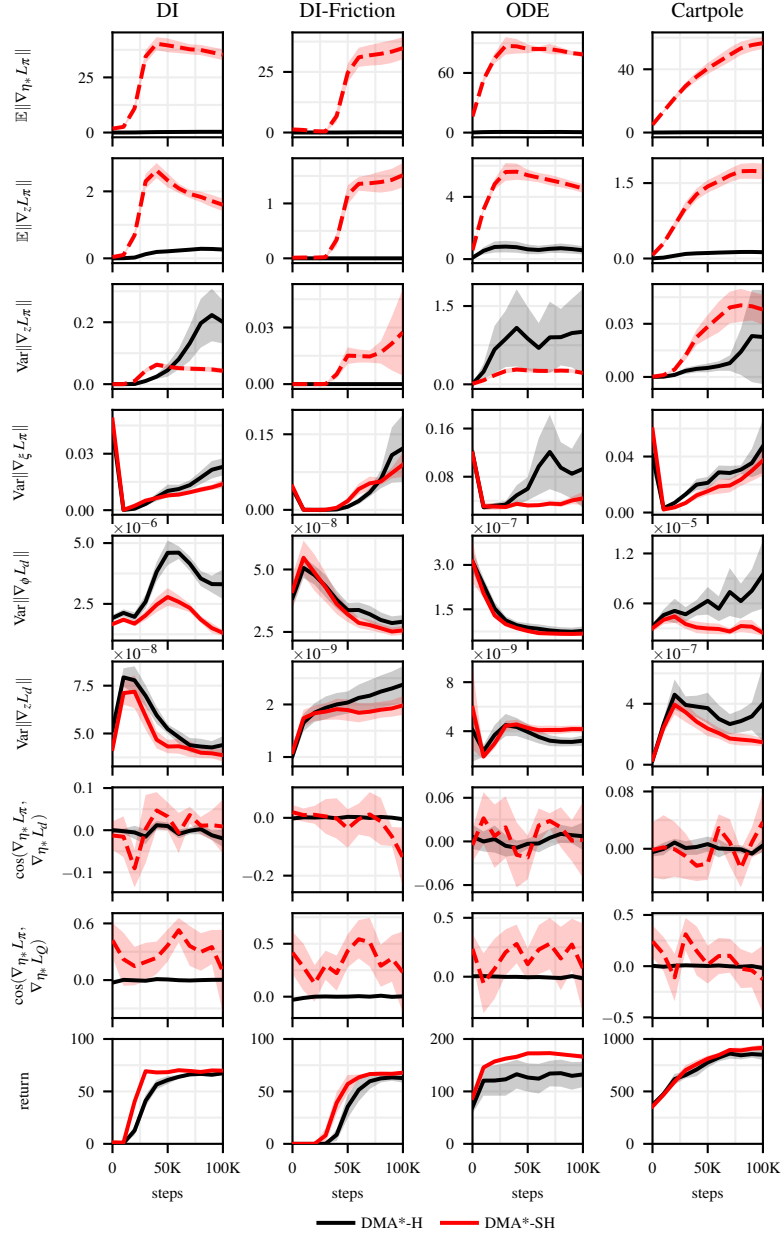


Figure 8: Gradient analysis comparing shared hypernetworks (DMA*-SH, red) vs. separate hypernetworks (DMA*-H, black) across DI, DI-Friction, ODE, and Cartpole environments. Dashed lines indicate quantities computed via a shadow graph obtained by temporarily removing the stop-gradient through the adapter pathway (e.g., $\nabla_{\eta} L_{\pi}$ or $\nabla_z L_{\pi}$ in DMA*-SH, where neither η nor z receives policy gradients during training due to detachment of ω). This enables hypothetical gradient evaluation without altering the training loop. Here, η_* denotes the relevant hypernetwork parameters: in shared, the single η (optimized solely via the dynamics loss L_d); in separate, the module-specific hypernetworks (e.g., η^{π} for policy-related gradients). L_d , L_{π} and L_Q correspond respectively to the FD loss, L_{ξ} (actor) and L_{ζ} (Q-function).

B Extended Related Work

Zero-shot generalization in contextual RL. Contextual RL has been studied from multiple perspectives, including CMDPs, domain randomization, and meta-RL [Hallak et al., 2015, Modi et al., 2018, Beck et al., 2023a]. The survey by Kirk et al. [2023] highlights its importance for zero-shot generalization, noting that separating training and evaluation context sets enables systematic evaluation. Broadly, two directions are distinguished: 1) explicit context is observable as privileged information [Chen et al., 2018, Seyed Ghasemipour et al., 2019, Ball et al., 2021, Eghbal-zadeh et al., 2021, Sodhani et al., 2021, Mu et al., 2022, Benjamins et al., 2023, Prasanna et al., 2024], and 2) context must be inferred implicitly from past experience [Chen et al., 2018, Xu et al., 2019, Lee et al., 2020, Seo et al., 2020, Xian et al., 2021, Sodhani et al., 2022, Melo, 2022, Evans et al., 2022, Ndir et al., 2024, Röder et al., 2025]. Our work follows the second approach, focusing on self-supervised context inference via dynamics-model alignment. Recurrent agents may also learn internal context representations [Grigsby et al., 2024a,b, Luo et al., 2024, Hafner et al., 2019, 2025], though these are typically not aligned with the underlying dynamics. Closely related, Beukman et al. [2023] utilize hypernetworks [Ha et al., 2017] to incorporate context into RL models. Our approach differs fundamentally: We do not assume access to explicit context, and rather than employing separate hypernetworks for the policy/Q-function, we train a single hypernetwork jointly with the dynamics model, which is then shared across policy and action-value networks.

Meta-RL. Meta-RL aims to enable agents to rapidly adapt to unseen tasks with minimal data [Beck et al., 2023a], often by learning policies that infer task structure from prior interactions, sometimes using hypernetworks [Beck et al., 2022, 2023b]. Most meta-RL approaches require fine-tuning on new tasks across multiple episode rollouts [Duan et al., 2016, Finn et al., 2017, Nagabandi et al., 2018, Rakelly et al., 2019, Zintgraf et al., 2019], which is incompatible with our zero-shot generalization setting. VariBAD [Zintgraf et al., 2020] and TrMRL [Melo, 2022] are not subject to this limitation, as they have been shown to adapt to the task within the first rollout. Recent advances in context-based offline meta-RL are also promising, as these algorithms aim to infer latent task information, both reward- and dynamics-based, from static datasets [Li et al., 2021, Dorfman et al., 2021, Yuan and Lu, 2022, Li et al., 2024a,b]. However, since these methods typically rely on assumptions inherent to the offline setting, it remains an open question whether they can be robustly applied to online RL [Li et al., 2024a].

Context in cognition. Beyond RL, cognitive modeling suggests that humans segment the environment into context-like events [Zacks and Tversky, 2001, Zacks et al., 2007, Butz, 2016]. For instance, the recurrent REPRIME model learns latent context representations from scratch, distinguishing dynamic regimes [Butz et al., 2019]. More recent work differentiates event segmentation from context inference, showing that contextual priors support learning of sensorimotor repertoires and memory structures [Heald et al., 2021, 2023]. Bayesian active inference models indicate that context can reduce computational effort while accurately modeling human behavior [Marković et al., 2021, Schwöbel et al., 2021, Butz, 2022, Cuevas Rivera and Kiebel, 2023, Parr et al., 2023, Mittenbühler et al., 2024]. In cognitive modeling-inspired deep learning, contextualized hypernetworks have been introduced in various forms, showing superior generalization and emergent compositionality [Sugita et al., 2011], the emergence of affordance maps [Scholz et al., 2022], as well as the possibility to focus object-oriented encoding pipelines [Traub et al., 2024]. At the intersection of neuroscience, developmental psychology, cognitive modeling, and machine learning, context inference and context-conditioned learning appear to be critical to enabling robust behavioral learning in complex environments [Butz et al., 2024].

C Algorithms

For DMA/DMA* we isolate encoder training to the dynamics objective by detaching z_t in RL losses (Algorithm 1); analogously, DMA*-SH detaches ω in RL losses (Algorithm 2). See Figure 1.

Algorithm 1: Training loop **DMA/DMA***

Require: Context set $\mathcal{C}_{\text{train}} = \{c_i\}_{i=1\dots n_c}$ sampled from context range for training, learning rates $\alpha_1, \alpha_2, \alpha_3$

- 1: Init. replay buffers \mathcal{B}^c for each context
- 2: Init. context windows (deque)
 $\tau_{t=0}^c = \{(s_{t-k}, a_{t-k}, \delta s_{t+1-k})\}_{k:1\dots K} \sim \pi_{\text{random}}$ for each context
- 3: **for** step in training steps **do**
- 4: *// Collect data in environment*
- 5: **for** c in $\mathcal{C}_{\text{train}}$ **do**
- 6: Encode past transitions $z_t = g_\phi(\tau_t^c)$
- 7:
- 8: Gather data from environment interaction with $a_t \sim \pi_\xi(\cdot|s_t, z_t)$
- 9: Add data to \mathcal{B}^c and update τ_t^c
- 10: **end for**
- 11: *// Training*
- 12: **for** c in $\mathcal{C}_{\text{train}}$ **do**
- 13: Sample RL batch $b^c \sim \mathcal{B}^c$ with corresponding context windows τ_t^c
- 14: Encode past transitions $z_t = g_\phi(\tau_t^c)$
- 15:
- 16: Predict $\delta \hat{s}_{t+1} = f_\theta(s_t, a_t, z_t)$
- 17: $L_\xi^c = L_\xi(\pi_\xi, b^c, z_t.\text{detach}())$
- 18: $L_\zeta^c = L_\zeta(Q_\zeta, b^c, z_t.\text{detach}())$
- 19: $L_{\phi, \theta}^c = \|\delta \hat{s}_{t+1} - \delta s_{t+1}\|_2^2$
- 20: **end for**
- 21: $\xi \leftarrow \xi - \alpha_1 \nabla_\xi \sum_c L_\xi^c$
- 22: $\zeta \leftarrow \zeta - \alpha_2 \nabla_\zeta \sum_c L_\zeta^c$
- 23: $\phi \leftarrow \phi - \alpha_3 \nabla_\phi \sum_c L_{\phi, \theta}^c$
- 24: $\theta \leftarrow \theta - \alpha_3 \nabla_\theta \sum_c L_{\phi, \theta}^c$
- 25:
- 26: **end for**

Algorithm 2: Training loop **DMA*-SH**

Require: Context set $\mathcal{C}_{\text{train}} = \{c_i\}_{i=1\dots n_c}$ sampled from context range for training, learning rates $\alpha_1, \alpha_2, \alpha_3$

- 1: Init. replay buffers \mathcal{B}^c for each context
- 2: Init. context windows (deque)
 $\tau_{t=0}^c = \{(s_{t-k}, a_{t-k}, \delta s_{t+1-k})\}_{k:1\dots K} \sim \pi_{\text{random}}$ for each context
- 3: **for** step in training steps **do**
- 4: *// Collect data in environment*
- 5: **for** c in $\mathcal{C}_{\text{train}}$ **do**
- 6: Encode past transitions $z_t = g_\phi(\tau_t^c)$
- 7: **Compute weights** $\omega = h_\eta(z_t)$
- 8: Gather data from environment interaction with $a_t \sim \pi_{\xi, \omega}(\cdot|s_t)$
- 9: Add data to \mathcal{B}^c and update τ_t^c
- 10: **end for**
- 11: *// Training*
- 12: **for** c in $\mathcal{C}_{\text{train}}$ **do**
- 13: Sample RL batch $b^c \sim \mathcal{B}^c$ with corresponding context windows τ_t^c
- 14: Encode past transitions $z_t = g_\phi(\tau_t^c)$
- 15: **Compute weights:** $\omega = h_\eta(z_t)$
- 16: Predict $\delta \hat{s}_{t+1} = f_{\theta, \omega}(s_t, a_t)$
- 17: $L_\xi^c = L_\xi(\pi_{\xi, \omega}.\text{detach}(), b^c)$
- 18: $L_\zeta^c = L_\zeta(Q_\zeta, \omega.\text{detach}(), b^c)$
- 19: $L_{\phi, \theta, \eta}^c = \|\delta \hat{s}_{t+1} - \delta s_{t+1}\|_2^2$
- 20: **end for**
- 21: $\xi \leftarrow \xi - \alpha_1 \nabla_\xi \sum_c L_\xi^c$
- 22: $\zeta \leftarrow \zeta - \alpha_2 \nabla_\zeta \sum_c L_\zeta^c$
- 23: $\phi \leftarrow \phi - \alpha_3 \nabla_\phi \sum_c L_{\phi, \theta, \eta}^c$
- 24: $\theta \leftarrow \theta - \alpha_3 \nabla_\theta \sum_c L_{\phi, \theta, \eta}^c$
- 25: $\eta \leftarrow \eta - \alpha_3 \nabla_\eta \sum_c L_{\phi, \theta, \eta}^c$
- 26: **end for**

D Hyperparameters and Implementation Details

Table 4 provides an overview of the used hyperparameters of the SAC agent, the context encoder, the dynamic model and the hypernetwork. We did not perform any tuning for SAC and kept hyperparameters standard as provided in CleanRL [Huang et al., 2022].

At the core of the context encoder, we use an LSTM layer whose final hidden state serves as the context representation z_t . This follows prior work employing MLPs, RNNs, or Transformer encoder layers as context encoders [Rakelly et al., 2019, Evans et al., 2022]. Prior work [Rakelly et al., 2019] used permutation-invariant encoders so that context inference does not depend on the temporal ordering of transitions within the context window. Motivated by this, we process a uniformly subsampled fraction of transitions from the context window in random order before feeding them to the LSTM. The context window size K depends on the environment: tasks derived from the DM Control Suite require a larger K than others. The context encoder samples a random 20% of the K transitions as input τ_t^c ; for example, in the DM Control Suite it observes only $128 \times 0.2 \approx 25$ transitions.

For our hypernetworks, we use the framework of von Oswald et al. [2020]. The adapter introduces a bottleneck, and importantly, we do not apply an activation function before it. Our design also allows the adapter to be bypassed via a skip connection. The design choices regarding the hypernetworks and adapters match those in DA [Beukman et al., 2023], where the placement of activation functions is likewise crucial. We reimplemented DA and verified that its performance is comparable to the original implementation.

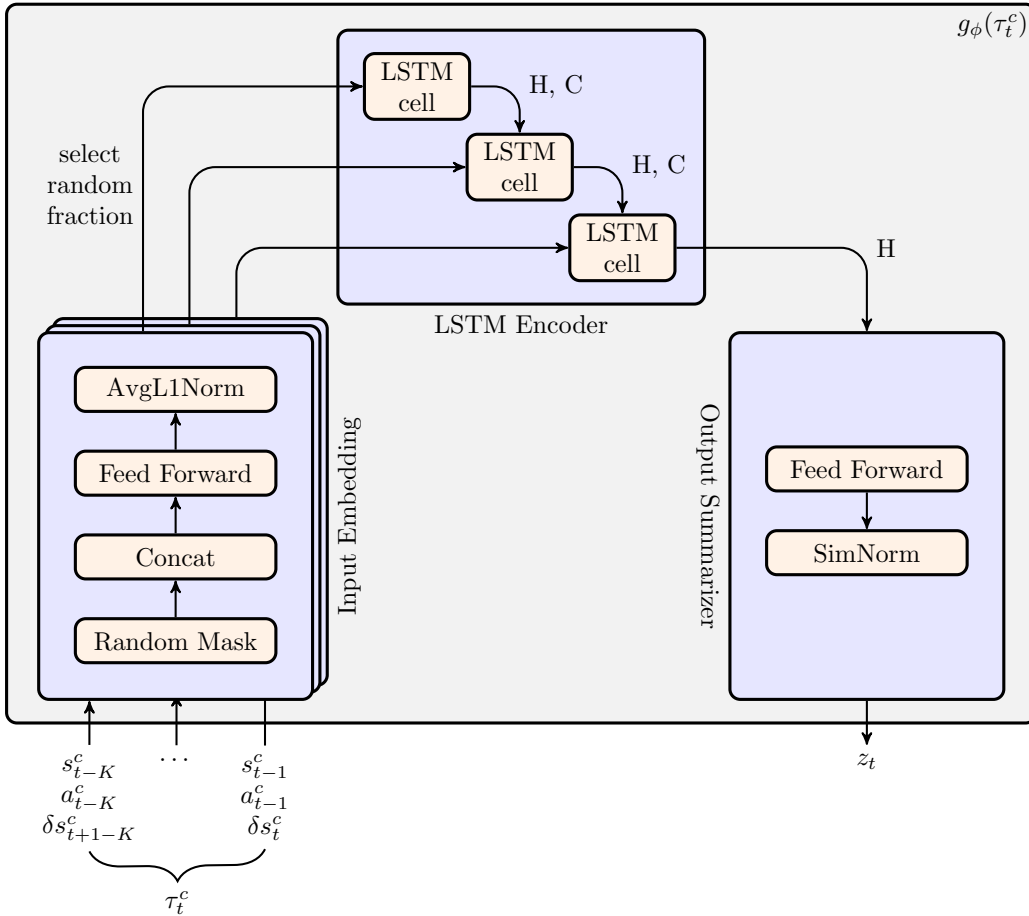


Figure 9: Architecture of the context encoder.

Table 4: Hyperparameters.

Module	Name	Value
SAC	Buffer capacity (per context)	100 000
	Batch size	256
	Discount γ	0.99
	Optimizer	Adam
	Critic LR	0.001
	Actor LR	0.0003
	Temperature LR	0.0003
	Critic soft target update τ	0.005
	Init temperature	1.0
	Hidden dims	(256, 256)
	Activation function	ReLU
Context encoder	LR	0.0003
	Model dim	32
	Context dim	8
	Context window size K (DI, ODE)	24
	Context window size K (DMC- and Gymnasium based)	128
	Context window fraction	0.2
	Context encoder type	LSTM
	LSTM representation state	hidden H
	Activation function	ReLU
	Input Norm	AvgL1Norm
	Output Norm	SimNorm
		Input Masking (DMA*)
	Input Masking (DMA*-SH)	0.4
Dynamic model	LR	0.0003
	Hidden dims	(256, 256)
	Activation function	ReLU
Hypernetwork	LR	0.0003
	Hidden dims	(64, 64)
	Activation function	ReLU
Adapter	Bottleneck	32
	Skip connection	True
	Pre adapter activation function	None
	Post adapter activation function	ReLU

E AIB: Discontinuous Context-to-Dynamics Environment Suite

This section gives concise descriptions of the environments summarized in Table 1, together with the context variables and our overlap classification (Definition A.8). Unless noted otherwise, each environment uses two context dimensions sampled once per episode and held fixed throughout. The context does not vary within an episode, it varies across episodes.

At a high level, environments are *overlapping* when a single robust context-unaware policy can often perform reasonably across the context range, as is typical for continuous physical parameters inducing gradual changes in optimal behavior. Environments are *non-overlapping* when context instances induce mutually incompatible optimal control laws. AIB creates such structure through discontinuous action-effect changes, including actuator inversion, actuator permutations, and continuous weakly non-overlapping dynamics. Actuator inversion (Definition A.10) is the minimal binary case: a context $c \in \{\pm 1\}$ multiplies the action before it affects the dynamics, $P^c(s'|s, a) = P(s'|s, c \cdot a)$. This makes identical actions induce opposite state transitions across modes. Actuator permutations similarly swap control channels, while ODE/ODE- k provide continuous contexts with empirically low policy overlap over the benchmark ranges.

E.1 Environments

Several tasks are taken from the DeepMind Control Suite (DMC) [Tassa et al., 2018] and Gymnasium MuJoCo [Towers et al., 2024]. The ODE/ODE- k environments follow prior work [Beukman et al., 2023]. The suite contains three types of non-overlap: binary sign inversions, actuator permutations, and continuous weakly non-overlapping dynamics.

DI (Custom; non-overlapping). We consider a custom two-dimensional double-integrator point-mass task. The state consists of planar position and velocity, and actions apply planar forces in the x and y directions. The agent is initialized away from the goal and receives a sparse reward: +1 on reaching the goal, 0 otherwise. Context variables are (i) the mass and (ii) an actuator inversion factor $c \in \{\pm 1\}$ that flips the sign of the applied force. The inversion produces incompatible control laws related approximately by a sign flip, yielding non-overlapping structure.

DI-Friction (Custom; overlapping). This environment matches DI but includes frictional damping. Context variables are (i) mass and (ii) friction coefficient. Both are continuous physical parameters and typically induce gradual shifts in the optimal control law. Robust context-unaware agents can often handle a wide range, so we treat this family as overlapping.

DI-Perm (Custom; non-overlapping). This environment considers an actuator permutation-variant of the DI. Context variables are (i) the mass and (ii) an actuator permutation factor $q \in \{0, 1\}$ that swaps the x - and y -components of the applied force. For $q = 0$, actions are applied normally; for $q = 1$, the force components are exchanged, i.e. (a_x, a_y) is applied as (a_y, a_x) . The permutation produces incompatible control laws related approximately by a coordinate permutation, yielding non-overlapping structure.

ODE (Custom; weakly non-overlapping). We include a scalar regulation task where the state evolves as

$$x_{t+1} = x_t + \dot{x}_t dt, \quad \dot{x}_t = c_1 a_t + c_2 a_t^2,$$

with context $c = (c_1, c_2) \in \mathbb{R}^2$. The objective is to choose a_t to keep x_t near zero. Although the context variables are continuous (no explicit binary inversion), the action-to-state map depends multiplicatively and nonlinearly on c . Regulating $x_t > 0$ toward zero requires $\dot{x}_t < 0$. To first order, this requires $c_1 a_t < 0$: when $c_1 > 0$, the agent must choose $a_t < 0$, whereas when $c_1 < 0$, the agent must choose $a_t > 0$. Thus, the required action direction flips with $\text{sign}(c_1)$. This acts as a continuous analog of actuator inversion induced by c_1 crossing 0 rather than an explicit binary flag. The quadratic term $c_2 a_t^2$ adds context-dependent magnitude constraints, e.g., when $c_2 > 0$, excessively large $|a|$ makes the quadratic term dominate and can prevent $\dot{x}_t < 0$ even if $c_1 a_t < 0$. The primary action direction, however, is still dictated by $\text{sign}(c_1)$. Empirically, Beukman et al. [2023] report poor performance of context-unaware baselines on this task over their benchmark ranges, consistent with low policy overlap; we therefore classify ODE as weakly non-overlapping.

ODE- k (Custom; weakly non-overlapping). This environment extends the ODE task to k context dimensions via a higher-order polynomial

$$x_{t+1} = x_t + \dot{x}_t dt, \quad \dot{x}_t = \sum_{j=1}^k c_j a_t^j,$$

with context $c = (c_1, \dots, c_k)$. In ODE- k , sign variation in odd-order coefficients (c_1, c_3, \dots) can enforce directional action flips analogous to the c_1 effect in ODE. Even with fixed signs, varying the coefficients can substantially alter the polynomial’s shape, introducing multiple local minima or divergent directions that can cause the optimal control strategy to shift abruptly between disparate regimes. This increased structural complexity reduces policy overlap. Empirically, context-unaware agents perform poorly in these settings [Beukman et al., 2023]. We therefore classify ODE- k as weakly non-overlapping.

Cartpole (DMC; non-overlapping). We use the cartpole-balance-v0 task, where the agent balances a pole by applying horizontal forces to its base. The environment is contextualized by (i) pole length and (ii) an actuator inversion factor $c \in \{\pm 1\}$ that flips the sign of the applied force. This creates a discontinuous action-effect change and yields non-overlapping structure.

Cheetah (DMC; non-overlapping). We use the cheetah-run-v0 task, where a planar biped is rewarded for moving forward. The environment is contextualized by (i) leg-length scaling, which significantly affects gait dynamics, and (ii) an actuator inversion factor $c \in \{\pm 1\}$ applied to action effects. The inversion produces incompatible gaits/control commands across modes, yielding non-overlapping structure.

Reacher (Easy/Hard) (DMC; non-overlapping). We use the reacher-easy/hard-v0 tasks (easy and hard variants), where a two-link planar arm is rewarded for reaching a target sphere (larger in the easy variant than in the hard variant). The environment is contextualized by (i) an arm-length scaling factor, which changes the kinematics and dynamics, and (ii) an actuator inversion factor $c \in \{\pm 1\}$ that flips the effect of actions. The inversion induces incompatible reaching behaviors across modes, so we classify these contexts as non-overlapping.

Reacher (Easy/Hard)-Perm (DMC; non-overlapping). These environments consider actuator permutation-variants of Reacher (Easy/Hard). Context variables are (i) arm-length scaling and (ii) an actuator permutation factor $q \in \{0, 1\}$ that exchanges the actuation of the two reacher links shoulder and wrist. For $q = 0$, actions are applied normally; for $q = 1$, the action dimensions are swapped. The permutation produces incompatible control laws, yielding non-overlapping structure.

BallInCup (DMC; overlapping). We use the ball_in_cup-catch-v0 task, where an actuated cup moves in the vertical plane to swing and catch a ball attached by a tendon. The reward signal is sparse, i.e., +1 if the ball is in the cup, 0 otherwise. The environment is contextualized by (i) the tendon length and (ii) gravity. Since both context variables are continuous physical parameters that typically induce gradual changes in the swing dynamics, robust policies often transfer across ranges; we therefore classify this family as overlapping.

Walker (DMC; overlapping). We use the walker-walk-v0 task, where a planar walker is rewarded for moving forward. The environment is contextualized by (i) actuator strength and (ii) gravity. Both are continuous parameters that induce smooth variations in locomotion difficulty and dynamics; we classify this family as overlapping.

WalkerGym (Gymnasium; overlapping). We use the Walker2d-v5 task, where a planar walker is rewarded for moving forward. The environment is contextualized by (i) actuator strength and (ii) gravity. Both are continuous and typically yield gradual policy variation; we classify this family as overlapping.

HopperGym (Gymnasium; overlapping). We use the Hopper-v5 task, where a planar hopper is rewarded for moving forward without falling over. The environment is contextualized by (i) actuator strength and (ii) gravity. Both are continuous and generally yield gradual variations in feasible hopping gaits; we classify this family as overlapping.

Table 5: **Context ranges and return bounds for AIB environments.** Each environment samples a context per episode, held fixed throughout. Contexts are two-dimensional for all environments (except ODE- k , which uses k coefficients with identical per-coefficient ranges to ODE). Columns Train, Eval-in, Eval-out specify the sampling supports for $\mathcal{C}_{\text{train}}$, $\mathcal{C}_{\text{eval-in}}$, and $\mathcal{C}_{\text{eval-out}}$, respectively; while supports may overlap, the sampled context instances are pairwise disjoint across sets. The last column gives return bounds $[J_E^{\text{lo}}, J_E^{\text{hi}}]$ used for score normalization.

Environment	Context	Context ranges			Training steps	Return bounds
		Train	Eval-in	Eval-out		
DI	mass actuator inversion	$[0.5, 1.5]$ $\{\pm 1\}$	$(0.5, 1.5)$ $\{\pm 1\}$	$[0.1, 0.5] \cup (1.5, 2.0)$ $\{\pm 1\}$	100 000	$[0, 100]$
DI-Friction	mass friction	$[0.5, 1.5]$ $[0.5, 1.5]$	$(0.5, 1.5)$ $(0.5, 1.5)$	$[0.1, 0.5] \cup (1.5, 2.0)$ $[0.1, 0.5] \cup (1.5, 2.0)$	100 000	$[0, 100]$
DI-Perm	mass actuator permutation	$[0.5, 1.5]$ $\{0, 1\}$	$(0.5, 1.5)$ $\{0, 1\}$	$[0.1, 0.5] \cup (1.5, 2.0)$ $\{0, 1\}$	100 000	$[0, 100]$
ODE	c_1 c_2	$[-5, 5]$ $[-5, 5]$	$(-5, 5)$ $(-5, 5)$	$[-10, -5] \cup (5, 10)$ $[-10, -5] \cup (5, 10)$	100 000	$[0, 200]$
ODE- k	all c_i ($i \in \{1, \dots, k\}$)	$[-5, 5]$	$(-5, 5)$	$[-10, -5] \cup (5, 10)$	100 000	$[0, 200]$
Cartpole	pole length actuator inversion	$[0.3, 0.85]$ $\{\pm 1\}$	$(0.3, 0.85)$ $\{\pm 1\}$	$[0.1, 0.3] \cup (0.85, 2.0]$ $\{\pm 1\}$	100 000	$[0, 1000]$
Cheetah	leg length actuator inversion	$[0.8, 1.2]$ $\{\pm 1\}$	$(0.8, 1.2)$ $\{\pm 1\}$	$[0.4, 0.8] \cup (1.2, 1.6]$ $\{\pm 1\}$	200 000	$[0, 1000]$
Reacher (E/H)	arm length actuator inversion	$[0.8, 1.2]$ $\{\pm 1\}$	$(0.8, 1.2)$ $\{\pm 1\}$	$[0.4, 0.8] \cup (1.2, 1.6]$ $\{\pm 1\}$	200 000	$[0, 1000]$
Reacher (E/H)-Perm	arm length actuator permutation	$[0.8, 1.2]$ $\{0, 1\}$	$(0.8, 1.2)$ $\{0, 1\}$	$[0.4, 0.8] \cup (1.2, 1.6]$ $\{0, 1\}$	200 000	$[0, 1000]$
BallInCup	gravity tendon length	$[8.0, 12.0]$ $[0.24, 0.36]$	$(8.0, 12.0)$ $(0.24, 0.36)$	$[1.0, 8.0] \cup (12.0, 20.0]$ $[0.1, 0.24] \cup (0.36, 0.5]$	200 000	$[0, 1000]$
Walker	gravity actuator strength	$[4.9, 14.7]$ $[0.5, 1.5]$	$(4.9, 14.7)$ $(0.5, 1.5)$	$[1.0, 4.9] \cup (14.7, 19.6]$ $[0.1, 0.5] \cup (1.5, 2.0]$	200 000	$[0, 1000]$
WalkerGym	gravity actuator strength	$[4.9, 14.7]$ $[0.5, 1.5]$	$(4.9, 14.7)$ $(0.5, 1.5)$	$[1.0, 4.9] \cup (14.7, 19.6]$ $[0.1, 0.5] \cup (1.5, 2.0]$	500 000	$[0, 5000]$
HopperGym	gravity actuator strength	$[4.9, 14.7]$ $[0.5, 1.5]$	$(4.9, 14.7)$ $(0.5, 1.5)$	$[1.0, 4.9] \cup (14.7, 19.6]$ $[0.1, 0.5] \cup (1.5, 2.0]$	500 000	$[0, 3800]$

E.2 Comparison with Existing Benchmarks

We distinguish AIB from two widely used contextual/multitask RL benchmarks: CARL [Benjamins et al., 2023] and Meta-World [Yu et al., 2020, McLean et al., 2025].

CARL contextualizes standard RL environments by exposing environment parameters as an explicit context (e.g., gravity, mass, friction, actuator strength) [Benjamins et al., 2023]. These variations are primarily continuous parameter shifts designed to preserve the environment’s intended semantics, and the benchmark does not target discontinuous context-to-dynamics changes in how actions affect state transitions.

Meta-World provides 50 qualitatively distinct robotic manipulation tasks (e.g., door opening, drawer closing, button pressing, object pushing) with parametric variation in goal and object positions within each task [Yu et al., 2020]. The benchmark evaluates whether agents can learn multiple skills and transfer to held-out tasks. Within any given task, the underlying transition dynamics remain fixed; only goals and object configurations change and the benchmark does not target context-induced dynamics shifts.

AIB still encompasses smooth physical parameter variations but complements these benchmarks by explicitly targeting discontinuous context-to-dynamics structure. In non-overlapping AIB settings, identical actions can induce incompatible effects across modes, e.g., opposite forces under inversion or swapped physical effects under actuator permutation, so optimal policies need not vary smoothly with context. Such discontinuities can arise in practice from coordinate-frame mismatches, mirrored joints, swapped motor polarities, swapped control channels, or interface inversions in teleoperation. See Remark A.9. Neither CARL nor Meta-World is designed to isolate such failure modes, making AIB a targeted *stress test* for zero-shot generalization under discontinuous context shifts.

F Ablations and Design Rationale

We briefly summarize the key design choices in DMA*-SH. Additional implementation details for the context encoder and hypernetwork appear in Appendix D. See Appendices A.3–A.5 for extended discussion of how these choices shape the learned context geometry.

Input masking. Hypernetworks can amplify small perturbations in z_t into large changes in ω . Random masking regularizes this pathway by encouraging g_ϕ to learn a redundant, distributed code that spreads information across the context window and across coordinates, so that $\omega = h_\eta(z_t)$ remains context-sensitive but less dependent on any single input component. A masking ratio of 40% performed best in our experiments, and performance is robust across a range of substantial masking ratios (Figure 11).

AvgL1Norm input normalization. After masking and a linear projection, we normalize per sample using AvgL1Norm. Among the input-normalization options we tested within the context window, including LayerNorm [Ba et al., 2016], AvgL1Norm [Fujimoto et al., 2023], SimNorm [Lavoie et al., 2023, Hansen et al., 2024], and WindowNorm, AvgL1Norm yielded the most reliable performance (Figure 12).

SimNorm output normalization. Normalizing the context embedding z_t is critical for stable online training. Among LayerNorm, AvgL1Norm, and SimNorm, the best performance in our experiments was achieved with SimNorm (Figure 13).

Adapter weight sharing. Sharing a single dynamics-trained hypernetwork with the policy and Q-function is more effective than training separate hypernetworks for the RL modules. An ablation using separate hypernetworks is presented in Figure 15. Also see Appendix A.5 and Figure 8 for an extended discussion.

We perform a range of ablations that motivate our design choices. For these ablations, results are aggregated across a subset of contextualized environments from the AIB benchmark (Section 5 and Appendix E): DI, DI-Friction, ODE, Cartpole, BallInCup, and Walker.

Figure 10 reports probability of improvement (PoI) for DMA* and DMA*-SH, following Agarwal et al. [2021]. PoI indicates how likely one design choice improves over another, but it does not quantify the magnitude of the improvement. In Figure 2 we compare vanilla DMA to DMA* and DMA*-SH, indicating that the design choices cumulatively have a significant impact.

In Figure 11 we compare IQM scores [Agarwal et al., 2021] for different ratios of random input masking of actions, states, and next-state differences in τ_t^c , suggesting that 20% is beneficial for DMA* and 40% is beneficial for DMA*-SH. Especially for DMA*-SH, substantial performance degradation appears only at relatively high masking ratios, indicating robustness to variation in the sampled context inputs τ_t^c .

In Figures 12 and 13 we compare different normalization variants for the input and output of the context encoder. The intuition about AvgL1Norm and SimNorm provided in Section 4.1 and in prior work [Fujimoto et al., 2023, Lavoie et al., 2023, Hansen et al., 2024] is also reflected in performance. Our dynamics-alignment loss (1), which operates on state differences, encourages the encoder to organize z_t according to relative differences between contexts rather than absolute magnitudes, consistent with a scale-insensitive perspective; see Appendix A.4 for discussion.

Figure 14 motivates the choice of window size $K = 24$ for DI, DI-Friction, and ODE, and $K = 128$ for the DMC- and Gymnasium-based environments. The impact of K appears modest once a sufficiently large minimum window size is used.

Overall, the ablations indicate that appropriate normalization is important for dynamics-aligned context encoders in zero-shot generalization for contextual RL, and that input masking can further improve performance. This is also reflected in Figure 15 for DMA*-SH.

Furthermore, Figure 15 shows that the shared hypernetwork design outperforms an architecture with separate hypernetworks for the dynamics model, policy, and Q-function (DMA*-H; Section A.5). To further isolate the contribution of dynamics alignment for RL, we constructed a variant DMA*-H (RL only) that applies a hypernetwork to the RL modules but not to the dynamics model, so the RL adapter weights are not dynamics-aligned. Apart from a KL-loss term and a contrastive-loss term, DMA*-H (RL only) closely mirrors the hypernetwork structure of R2PGO [Li et al., 2024b] in an

online RL setting. Allowing actor gradients to propagate directly through the shared hypernetwork leads to near-zero returns (DMA*-SH, actor non-detached). This shows that letting RL rewrite the shared ω is harmful, and that gradient detachment is necessary; see Appendix A.6. Figure 15 also shows that normalization, masking, hypernetwork sharing, and dynamics-model alignment are all beneficial; see Appendix A.4 for a detailed discussion.

For the unaware, recurrent baseline Amago-v2 [Grigsby et al., 2024b], we use a GRU trajectory encoder instead of the default Transformer variant. This is justified by overall slightly stronger performance (Figure 16).

PEARL [Rakelly et al., 2019] is used as a baseline to compare against DMA* and DMA*-SH. PEARL was originally proposed for meta-RL settings where the latent context captures task variation via rewards, and the context encoder is trained through RL losses. To better match our dynamics-aligned setting, we consider a variant that couples PEARL’s probabilistic context encoder to a dynamics model, which we denote as DMA-Pearl. A comparison of PEARL and DMA-Pearl is provided in Figure 17a. Furthermore, Figure 17b reports performance for different choices of β that weight the KL regularization in PEARL. DMA-Pearl improves over vanilla DMA (Table 2), highlighting the benefits of a probabilistic context encoder with KL regularization in this setting. However, integrating these design elements into DMA* and DMA*-SH does not yield further gains (Figure 17a).

We also conducted experiments with VariBAD [Zintgraf et al., 2020], testing two different KL-weights β . While it achieves comparable performance in the overlapping DI-Friction setting, it struggles considerably with the non-overlapping contextualizations in DI and ODE. For this reason, and given that we already include DMA-Pearl in our comparisons, we exclude VariBAD as a baseline in the remainder of the paper (Section 6.2).

Methods built on smooth latent dynamics priors such as VariBAD and DMA-Pearl can struggle when their objectives explicitly encourage latent embeddings to vary continuously with respect to context trajectories. This inductive bias can be mismatched to tasks whose true context-to-dynamics map exhibits genuine discontinuities (as in DI with actuator inversion), where the correct representation requires a sign flip rather than a smooth interpolation (Remark A.13). In contrast, DMA*-SH can represent such discontinuities through multiplicative modulation in the hypernetwork pathway, which allows sharp directional changes in the induced policy and critic without an explicit ELBO or latent-prior penalty. As a result, DMA*-SH avoids the continuity bias inherent to latent-prior methods and can realize the functional geometry required for discontinuous contextual RL.

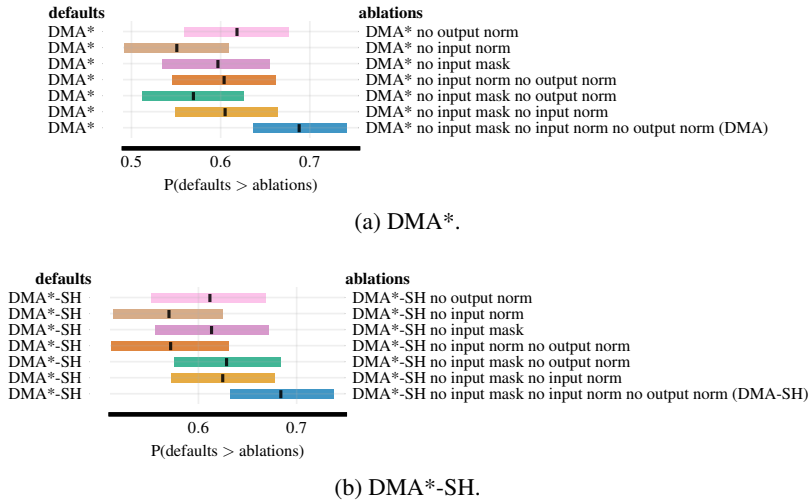


Figure 10: Probability of improvement (POI) [Agarwal et al., 2021] based on AER scores (Section 6.1) aggregated over six contextualized environments and over contexts in the three context sets $\mathcal{C}_{\text{train}}$, $\mathcal{C}_{\text{eval-in}}$ and $\mathcal{C}_{\text{eval-out}}$. For the proposed DMA* and DMA*-SH, we ablate separately the random masking, input and output normalization, a combination of two, or everything at once (DMA, DMA-SH).

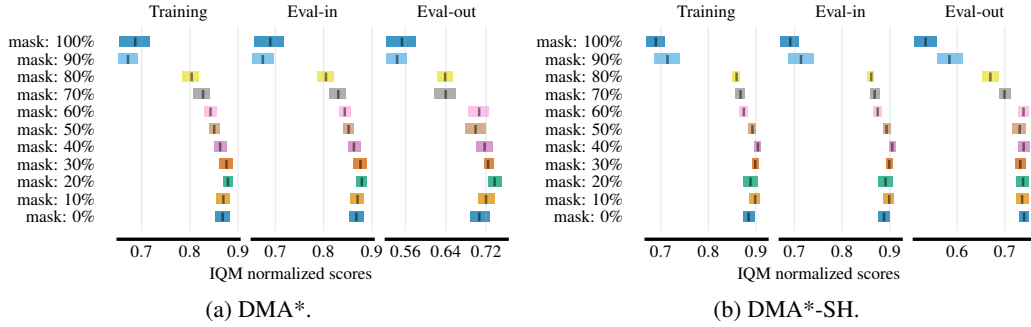


Figure 11: Interquartile mean (IQM) [Agarwal et al., 2021] based on AER scores (Section 6.1) aggregated over six contextualized environments. We distinguish results for contexts in the three context sets C_{train} , $C_{\text{eval-in}}$ and $C_{\text{eval-out}}$. We compare different ratios for the random input masking. When averaging over the three context sets, best performance is achieved using a ratio of 20% for DMA* and 40% for DMA*-SH.

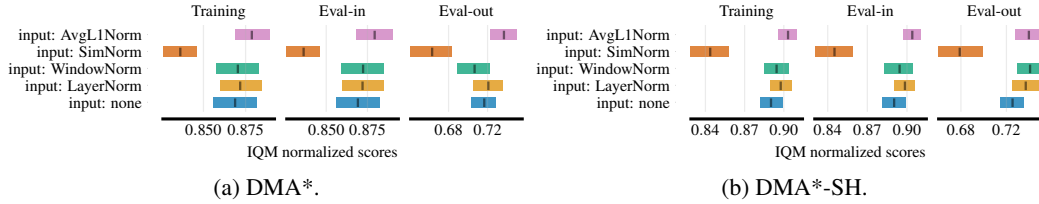


Figure 12: Interquartile mean (IQM) [Agarwal et al., 2021] based on AER scores (Section 6.1) aggregated over six contextualized environments. We distinguish results for contexts in the three context sets C_{train} , $C_{\text{eval-in}}$ and $C_{\text{eval-out}}$. We compare different types of input normalization. When averaging over the three context sets, best performance is achieved using AvgL1Norm in both DMA* and DMA*-SH.

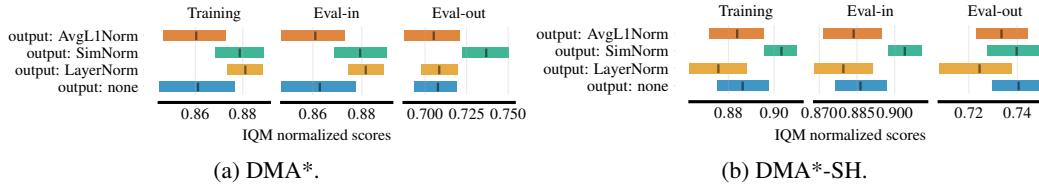


Figure 13: Interquartile mean (IQM) [Agarwal et al., 2021] based on AER scores (Section 6.1) aggregated over six contextualized environments. We distinguish results for contexts in the three context sets C_{train} , $C_{\text{eval-in}}$ and $C_{\text{eval-out}}$. We compare different types of output normalization. When averaging over the three context sets, best performance is achieved using SimNorm in both DMA* and DMA*-SH.

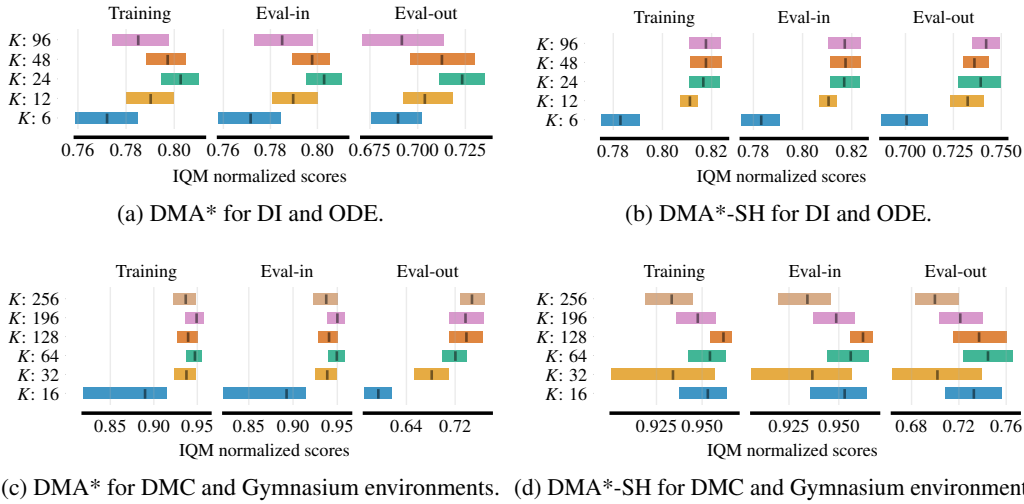


Figure 14: Interquartile mean (IQM) comparing different context window sizes justifying the choice of 24 for DI and ODE environments and 128 for DMC- and Gymnasium-based environments.

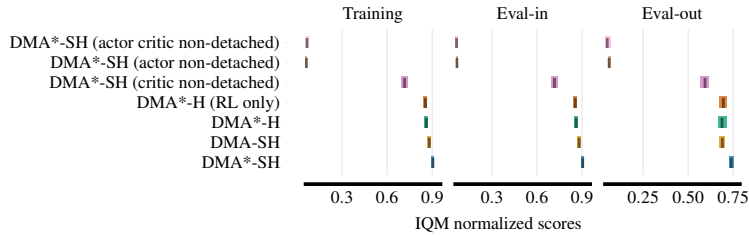


Figure 15: Interquartile mean (IQM) [Agarwal et al., 2021] based on AER scores (Section 6.1) aggregated over six contextualized environments. We distinguish results for contexts in the three context sets $\mathcal{C}_{\text{train}}$, $\mathcal{C}_{\text{eval-in}}$, and $\mathcal{C}_{\text{eval-out}}$. We compare DMA*-SH to a variant without normalization and masking (DMA-SH) and to an architecture that does not share the hypernetwork (DMA*-H). Instead, DMA*-H uses separate hypernetworks for the dynamics model, policy, and Q-value function. Apart from a KL-loss term and a contrastive-loss term, DMA*-H (RL only) closely resembles R2PGO [Li et al., 2024b] in an online RL setting. It does not employ a hypernetwork for the dynamics model, so the adapter weights for the RL modules are not aligned with the dynamics model. We also ablate gradient flow through the actor and/or critic pathways, denoted DMA*-SH (actor/critic non-detached), by allowing RL gradients to update the shared hypernetwork. Results show that detachment is critical, especially in non-overlapping settings; see Appendix A.6. Our results show that normalization, masking, hypernetwork sharing, and dynamics-model alignment are all beneficial. For a detailed discussion, see Appendix A.4.

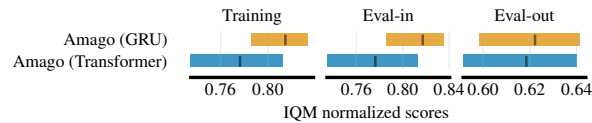


Figure 16: Interquartile mean (IQM) [Agarwal et al., 2021] based on AER scores (Section 6.1) aggregated over six contextualized environments. We distinguish results for contexts in the three context sets $\mathcal{C}_{\text{train}}$, $\mathcal{C}_{\text{eval-in}}$, and $\mathcal{C}_{\text{eval-out}}$. For Amago, we compare GRU- and Transformer-based trajectory encoders, justifying the use of the GRU variant.

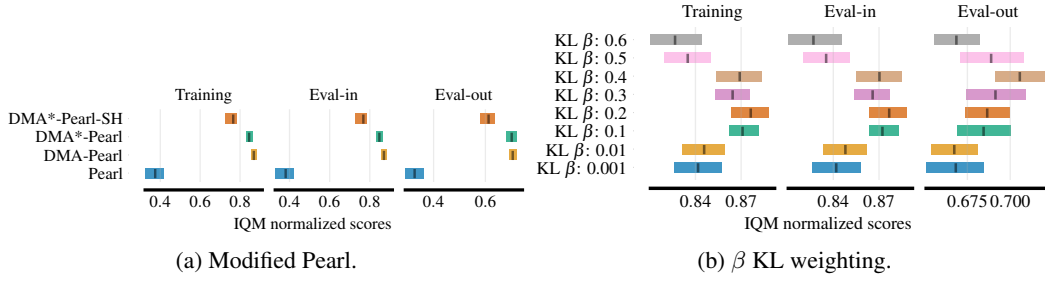


Figure 17: Interquartile mean (IQM) [Agarwal et al., 2021] based on AER scores (Section 6.1) aggregated over six contextualized environments. We distinguish results for contexts in the three context sets $\mathcal{C}_{\text{train}}$, $\mathcal{C}_{\text{eval-in}}$ and $\mathcal{C}_{\text{eval-out}}$. In a) we compare the original Pearl approach aligned with the Q-function to the dynamic model-aligned variant that we are using as a baseline, DMA-Pearl. Additionally, we incorporate our additions to DMA and the shared hypernetwork context utilization to Pearl. In b) we test different β weighting parameters for the KL term in Pearl and decided for $\beta = 0.4$ when using DMA-Pearl as a baseline.

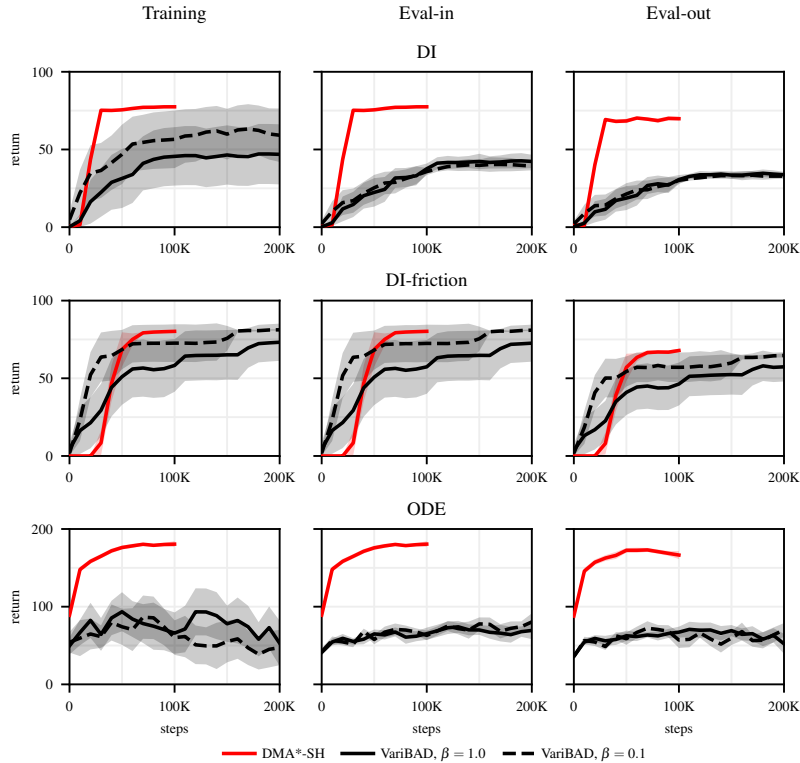


Figure 18: Returns over steps, averaged over the contexts used for training and evaluation, respectively. Comparison to the meta-RL approach VariBAD [Zintgraf et al., 2020]. See Remark A.13. VariBAD is based on the on-policy PPO, hence we allow for more environment steps. We do not observe any improvement after 200K steps. Two KL-weights (β) are tested.

G Detailed Results

G.1 Separate Result Tables, Learning Curves and Performance Gains

Table 2 aggregates AER scores over the three context sets. For a more detailed view, Tables 6–8 report results separately on $\mathcal{C}_{\text{train}}$, $\mathcal{C}_{\text{eval-in}}$, and $\mathcal{C}_{\text{eval-out}}$. DMA*-SH performs favorably across all three splits.

Figure 19 presents learning curves for six environments, shown separately for each context set. Depending on the environment, we run 100 000–500 000 gradient update steps and, for each context instance, the same number of environment steps. Since we train with $n_c = 20$ contexts, this corresponds to 2 000 000–10 000 000 total environment steps (Table 5). DMA*-SH exhibits consistently strong performance across these settings.

Table 6: AER scores with 95% confidence intervals for all environments. Results for the **context set** $\mathcal{C}_{\text{train}}$, comparing DMA*, DMA*-SH, and all baselines. Best AER scores are in bold; if multiple methods are highlighted for an environment, their differences are not distinguishable under the probability of improvement with 95% confidence intervals [Agarwal et al., 2021]. *All* aggregates all environments, *Overlap* aggregates those with overlapping contexts, and *Non-overlap* aggregates those with non-overlapping contexts. For the rows *All*, *Overlap*, and *Non-overlap*, scores are aggregated after environment-wise min–max scaling using environment-specific return bounds (Table 5).

Name	Context-Aware		Context-Unaware			Context-Inferred			
	Concat	DA	DR	Amago	DMA	DMA-Pearl	DMA*	DMA*-SH	
DI	75 [72, 77]	78 [77, 78]	17 [8, 25]	66 [55, 75]	74 [73, 75]	73 [71, 74]	77 [76, 78]	78 [78, 79]	
DI-Friction	71 [55, 79]	80 [79, 80]	72 [56, 81]	82 [81, 82]	62 [45, 74]	79 [78, 80]	71 [55, 80]	81 [80, 81]	
DI-Perm	77 [76, 78]	78 [77, 79]	54 [43, 63]	77 [71, 81]	76 [76, 77]	75 [74, 76]	76 [74, 77]	78 [77, 79]	
ODE	180 [175, 184]	183 [179, 188]	63 [54, 73]	178 [176, 180]	173 [170, 176]	174 [167, 181]	179 [175, 183]	183 [179, 186]	
Cartpole	929 [912, 945]	934 [902, 960]	658 [611, 706]	667 [586, 755]	919 [892, 944]	904 [854, 943]	941 [919, 961]	972 [959, 981]	
Cheetah	437 [418, 457]	452 [438, 469]	319 [293, 346]	488 [475, 505]	461 [429, 491]	428 [403, 458]	463 [443, 485]	484 [462, 504]	
Reacher (E)	923 [887, 954]	916 [870, 955]	584 [532, 639]	942 [929, 954]	903 [861, 939]	912 [877, 944]	925 [898, 950]	934 [917, 950]	
Reacher (E)-Perm	916 [883, 946]	899 [857, 932]	691 [655, 731]	910 [888, 930]	921 [888, 952]	929 [900, 954]	902 [875, 929]	933 [909, 954]	
Reacher (H)	724 [606, 823]	756 [704, 811]	290 [210, 381]	913 [897, 927]	700 [616, 780]	735 [649, 801]	743 [650, 825]	849 [773, 908]	
Reacher (H)-Perm	768 [666, 859]	804 [762, 849]	363 [261, 468]	807 [753, 852]	716 [604, 799]	731 [676, 793]	727 [623, 813]	895 [851, 934]	
BallInCup	976 [973, 977]	973 [971, 974]	961 [940, 974]	728 [577, 868]	975 [974, 977]	977 [975, 978]	975 [973, 976]	975 [973, 977]	
Walker	900 [890, 907]	882 [864, 898]	907 [898, 916]	845 [831, 857]	865 [812, 898]	908 [901, 915]	885 [866, 898]	915 [905, 925]	
WalkerGym	3068 [2751, 3355]	3834 [3663, 4030]	3171 [2946, 3400]	4551 [4217, 4894]	3502 [3264, 3733]	3571 [3282, 3876]	3333 [2960, 3718]	3726 [3606, 3820]	
HopperGym	2847 [2740, 2947]	2851 [2778, 2926]	2664 [2590, 2737]	2792 [2704, 2877]	2939 [2845, 3026]	2938 [2864, 3003]	2956 [2895, 3024]	2853 [2829, 2879]	
All	0.79 [0.77, 0.81]	0.81 [0.81, 0.82]	0.56 [0.54, 0.58]	0.79 [0.78, 0.8]	0.78 [0.76, 0.8]	0.8 [0.79, 0.8]	0.8 [0.78, 0.81]	0.84 [0.83, 0.84]	
Overlap	0.79 [0.75, 0.82]	0.83 [0.82, 0.84]	0.79 [0.75, 0.81]	0.81 [0.78, 0.84]	0.79 [0.75, 0.82]	0.83 [0.82, 0.85]	0.8 [0.76, 0.83]	0.84 [0.83, 0.84]	
Non-overlap	0.79 [0.77, 0.81]	0.8 [0.79, 0.81]	0.44 [0.41, 0.46]	0.78 [0.77, 0.8]	0.78 [0.75, 0.8]	0.78 [0.77, 0.79]	0.79 [0.77, 0.81]	0.84 [0.83, 0.85]	

Table 7: AER scores with 95% confidence intervals for all environments. Results for the **context set** $\mathcal{C}_{\text{eval-in}}$, comparing DMA*, DMA*-SH, and all baselines. Best AER scores are in bold; if multiple methods are highlighted for an environment, their differences are not distinguishable under the probability of improvement with 95% confidence intervals [Agarwal et al., 2021]. *All* aggregates all environments, *Overlap* aggregates those with overlapping contexts, and *Non-overlap* aggregates those with non-overlapping contexts. For the rows *All*, *Overlap*, and *Non-overlap*, scores are aggregated after environment-wise min–max scaling using environment-specific return bounds (Table 5).

Name	Context-Aware		Context-Unaware			Context-Inferred			
	Concat	DA	DR	Amago	DMA	DMA-Pearl	DMA*	DMA*-SH	
DI	75 [72, 77]	78 [77, 78]	17 [8, 26]	66 [56, 75]	74 [73, 75]	73 [72, 74]	77 [76, 78]	78 [78, 79]	
DI-Friction	71 [55, 79]	80 [79, 80]	72 [56, 81]	82 [81, 82]	62 [46, 75]	79 [78, 80]	71 [55, 80]	81 [80, 81]	
DI-Perm	77 [76, 78]	78 [77, 79]	54 [43, 63]	77 [71, 81]	76 [76, 77]	75 [74, 76]	76 [74, 77]	78 [77, 79]	
ODE	181 [175, 185]	183 [178, 187]	63 [54, 73]	178 [176, 180]	173 [170, 176]	173 [166, 180]	178 [174, 183]	182 [179, 185]	
Cartpole	930 [912, 945]	935 [905, 959]	659 [611, 706]	668 [589, 752]	919 [894, 945]	905 [855, 943]	941 [919, 961]	972 [961, 981]	
Cheetah	440 [423, 455]	457 [444, 472]	319 [295, 347]	488 [471, 504]	463 [436, 493]	430 [403, 460]	461 [441, 483]	487 [466, 507]	
Reacher (E)	924 [887, 954]	917 [869, 954]	582 [529, 636]	942 [930, 953]	905 [864, 941]	913 [878, 946]	925 [897, 951]	931 [916, 947]	
Reacher (E)-Perm	916 [881, 945]	900 [857, 933]	692 [655, 734]	915 [897, 932]	922 [888, 950]	929 [899, 955]	902 [873, 930]	932 [906, 953]	
Reacher (H)	724 [613, 822]	757 [705, 818]	291 [209, 380]	916 [901, 928]	702 [620, 787]	736 [648, 804]	745 [654, 825]	848 [772, 905]	
Reacher (H)-Perm	769 [672, 860]	805 [762, 853]	364 [266, 467]	812 [755, 856]	715 [599, 801]	732 [680, 792]	728 [622, 821]	894 [848, 932]	
BallInCup	976 [974, 978]	974 [972, 975]	957 [924, 975]	731 [586, 868]	976 [974, 977]	977 [975, 979]	975 [973, 977]	976 [974, 977]	
Walker	904 [894, 911]	887 [865, 903]	909 [901, 916]	849 [836, 861]	866 [813, 901]	912 [905, 919]	888 [867, 903]	918 [909, 926]	
WalkerGym	3093 [2760, 3400]	3845 [3671, 4048]	3204 [2985, 3424]	4601 [4273, 4959]	3519 [3279, 3764]	3585 [3303, 3878]	3377 [2986, 3748]	3753 [3619, 3876]	
HopperGym	2844 [2733, 2942]	2857 [2777, 2944]	2673 [2610, 2736]	2795 [2706, 2883]	2947 [2845, 3039]	2954 [2886, 3016]	2976 [2912, 3047]	2853 [2830, 2875]	
All	0.79 [0.77, 0.81]	0.82 [0.81, 0.82]	0.56 [0.54, 0.59]	0.79 [0.78, 0.81]	0.78 [0.76, 0.8]	0.8 [0.79, 0.81]	0.8 [0.78, 0.81]	0.84 [0.83, 0.84]	
Overlap	0.79 [0.75, 0.82]	0.84 [0.82, 0.85]	0.79 [0.75, 0.81]	0.81 [0.78, 0.84]	0.79 [0.75, 0.82]	0.83 [0.82, 0.85]	0.81 [0.76, 0.84]	0.84 [0.84, 0.84]	
Non-overlap	0.79 [0.77, 0.81]	0.8 [0.79, 0.81]	0.44 [0.41, 0.46]	0.78 [0.77, 0.8]	0.78 [0.76, 0.8]	0.78 [0.77, 0.79]	0.79 [0.77, 0.81]	0.84 [0.83, 0.85]	

Table 8: AER scores with 95% confidence intervals for all environments. Results for the **context set** $C_{\text{eval-out}}$, comparing DMA*, DMA*-SH, and all baselines. Best AER scores are in bold; if multiple methods are highlighted for an environment, their differences are not distinguishable under the probability of improvement with 95% confidence intervals [Agarwal et al., 2021]. All aggregates all environments, *Overlap* aggregates those with overlapping contexts, and *Non-overlap* aggregates those with non-overlapping contexts. For the rows *All*, *Overlap*, and *Non-overlap*, scores are aggregated after environment-wise min-max scaling using environment-specific return bounds (Table 5).

Name	Context-Aware		Context-Unaware				Context-Inferred			
	Concat	DA	DR	Amago	DMA	DMA-Pearl	DMA*	DMA*-SH		
DI	65 [62, 68]	70 [69, 71]	16 [11, 21]	52 [44, 59]	42 [39, 46]	58 [55, 62]	70 [69, 71]	71 [70, 73]		
DI-Friction	54 [39, 64]	68 [65, 70]	61 [47, 70]	73 [72, 74]	45 [33, 53]	65 [62, 68]	62 [47, 70]	69 [67, 70]		
DI-Perm	65 [61, 68]	71 [69, 73]	43 [34, 50]	63 [57, 68]	50 [47, 55]	61 [58, 65]	64 [61, 67]	69 [65, 72]		
ODE	126 [118, 134]	172 [165, 179]	63 [54, 71]	148 [147, 148]	152 [148, 157]	165 [160, 169]	168 [162, 175]	173 [170, 176]		
Cartpole	731 [700, 762]	808 [754, 862]	613 [564, 660]	581 [529, 637]	862 [842, 880]	842 [800, 874]	901 [873, 930]	958 [941, 972]		
Cheetah	279 [264, 297]	241 [226, 255]	205 [189, 222]	268 [255, 283]	239 [221, 258]	208 [195, 223]	225 [212, 239]	252 [232, 272]		
Reacher (E)	824 [779, 872]	801 [748, 851]	526 [478, 577]	858 [842, 873]	804 [772, 836]	799 [767, 832]	833 [809, 859]	829 [802, 858]		
Reacher (E)-Perm	817 [782, 845]	801 [746, 845]	593 [558, 634]	761 [724, 798]	800 [758, 841]	818 [787, 844]	802 [768, 833]	843 [823, 867]		
Reacher (H)	601 [506, 692]	627 [591, 665]	217 [140, 305]	729 [692, 766]	561 [483, 635]	588 [515, 644]	603 [523, 668]	719 [674, 763]		
Reacher (H)-Perm	600 [531, 665]	638 [604, 673]	257 [201, 313]	598 [554, 640]	573 [467, 660]	541 [466, 619]	581 [488, 664]	731 [690, 773]		
BallInCup	821 [792, 846]	691 [659, 730]	667 [611, 714]	493 [433, 558]	784 [760, 811]	756 [733, 782]	749 [705, 790]	719 [684, 755]		
Walker	546 [521, 571]	582 [561, 605]	578 [557, 598]	565 [551, 578]	557 [531, 575]	591 [577, 605]	566 [551, 581]	585 [572, 598]		
WalkerGym	1944 [1773, 2158]	2283 [2163, 2422]	2058 [1954, 2162]	2619 [2488, 2757]	2285 [2177, 2397]	2329 [2162, 2518]	2170 [1985, 2344]	2296 [2186, 2389]		
HopperGym	1872 [1795, 1959]	1918 [1866, 1969]	1795 [1737, 1853]	1917 [1845, 1989]	2001 [1948, 2045]	2046 [2003, 2086]	1973 [1917, 2026]	1982 [1935, 2024]		
All	0.61 [0.6, 0.63]	0.65 [0.63, 0.66]	0.43 [0.41, 0.45]	0.61 [0.6, 0.61]	0.59 [0.57, 0.61]	0.63 [0.62, 0.64]	0.64 [0.63, 0.65]	0.68 [0.68, 0.69]		
Overlap	0.56 [0.52, 0.59]	0.58 [0.57, 0.6]	0.55 [0.51, 0.57]	0.56 [0.55, 0.58]	0.55 [0.53, 0.57]	0.6 [0.59, 0.61]	0.58 [0.54, 0.6]	0.59 [0.59, 0.6]		
Non-overlap	0.64 [0.63, 0.66]	0.69 [0.67, 0.71]	0.37 [0.35, 0.39]	0.63 [0.62, 0.64]	0.61 [0.59, 0.64]	0.65 [0.64, 0.66]	0.68 [0.66, 0.7]	0.73 [0.72, 0.74]		

Table 9: Relative performance gains (+%) of DMA*-SH compared to DR, Concat, and DA across regimes and environment types. The *Aggregated* regime considers results aggregated across context sets C_{train} , $C_{\text{eval-in}}$, and $C_{\text{eval-out}}$ (Table 2). The *Train* regime considers results for the context set C_{train} (Table 6). The *Eval-out* regime considers results for the context set $C_{\text{eval-out}}$ (Table 8). Relative gain is computed as $(\text{DMA*SH} - \text{baseline}) / \text{baseline} \times 100$. Values are based on AER scores aggregated across environment types (see last three columns *All*, *Overlap*, and *Non-overlap* in the respective Tables 2, 6 and 8).

Regime	Type	vs DR	vs Concat	vs DA
Train	All	50.0%	6.3%	3.7%
	Overlap	6.3%	6.3%	1.2%
	Non-overlap	90.9%	6.3%	5.0%
Eval-out	All	58.1%	11.5%	4.6%
	Overlap	7.3%	5.4%	1.7%
	Non-overlap	97.3%	14.1%	5.8%
Aggregated	All	51.9%	8.2%	3.9%
	Overlap	7.0%	7.0%	1.3%
	Non-overlap	95.1%	8.1%	3.9%

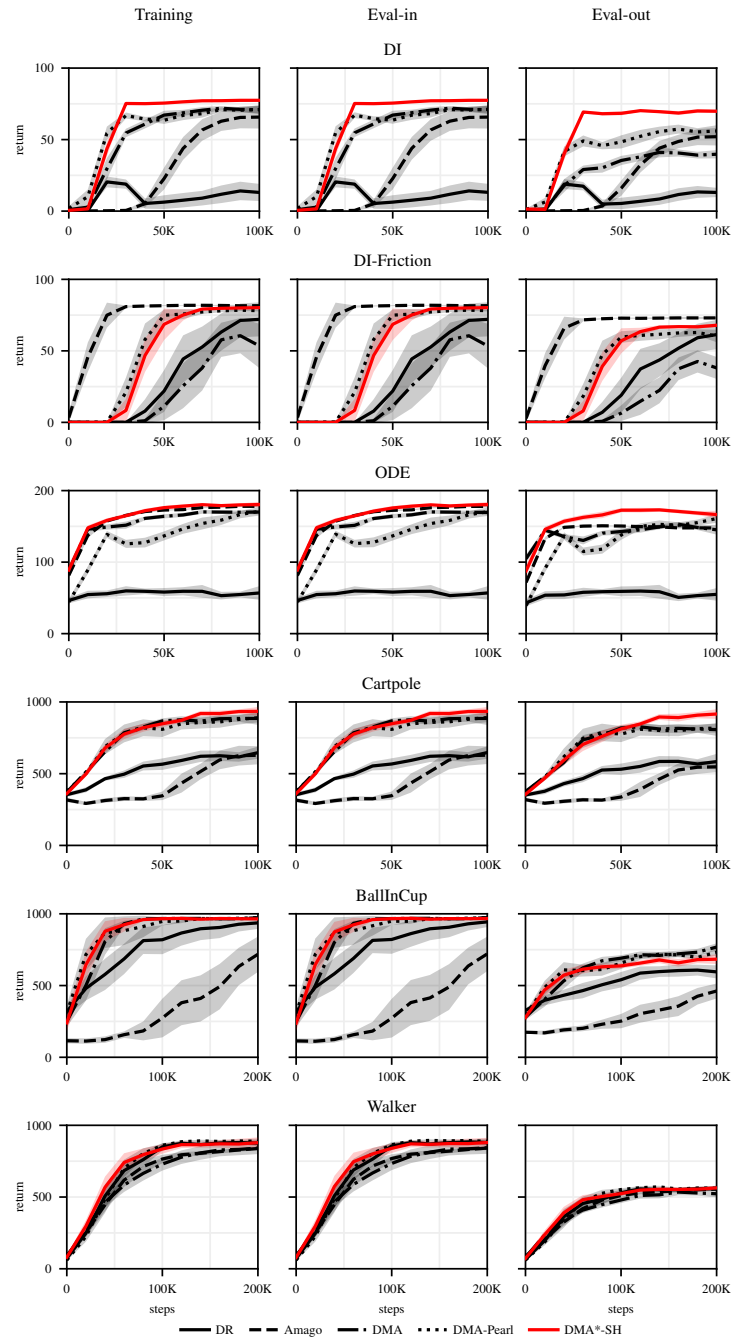


Figure 19: Returns over steps, averaged over the contexts used for training and evaluation, respectively. Comparison to baselines where context information is not explicitly available (Section 6.2).

G.2 Context-Instance Generalization Analysis

While the aggregated IQM provide a convenient high-level summary of performance across environments and context sets, contextual RL introduces an additional axis of variation that requires finer granularity [Benjamins et al., 2023, Ndir et al., 2024, Prasanna et al., 2024]. To make generalization behavior explicit, we complement the aggregated metrics with detailed visualizations at the level of individual context instances.

For each environment, we evaluate our proposed DMA*-SH against baselines across the full grid of training and evaluation contexts and visualize the results using context-wise bar plots (Figures 20–21) and heatmaps (Figures 22–27). These plots reveal how performance changes as evaluation contexts drift from the training distribution and highlight failure cases, such as the inability of the context-unaware DR baseline to handle non-overlapping dynamics (e.g., DI), or the challenges faced by the context-aware Concat method when dealing with extreme values of the context c_0 in the out-of-distribution regime of the ODE environment (Figure 24 and Table 8), indicating difficulties with both positive and negative values of c_0 . In Cartpole, DMA*-SH achieves impressive consistency across all context instances (Figure 21). This instance-level view exposes trends that aggregate statistics may obscure and provides a clearer understanding of where and how generalization breaks down.

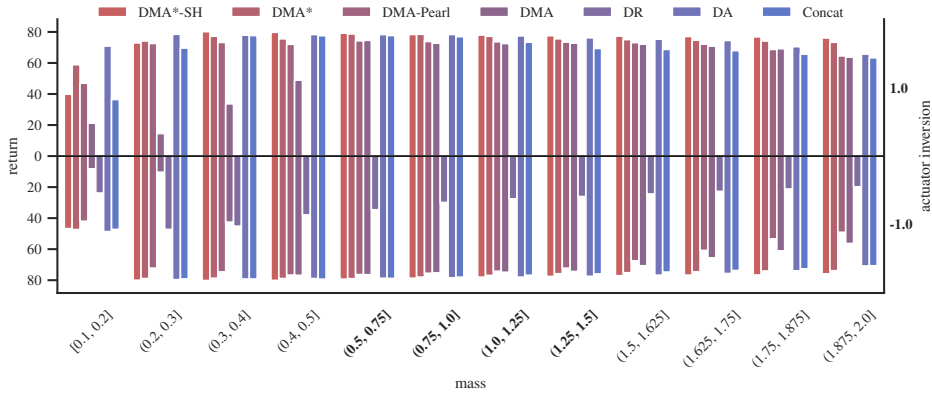


Figure 20: Bar plot for DI to visualize AER for individual context instances and different methods. Bold labels refer to contexts used during training.

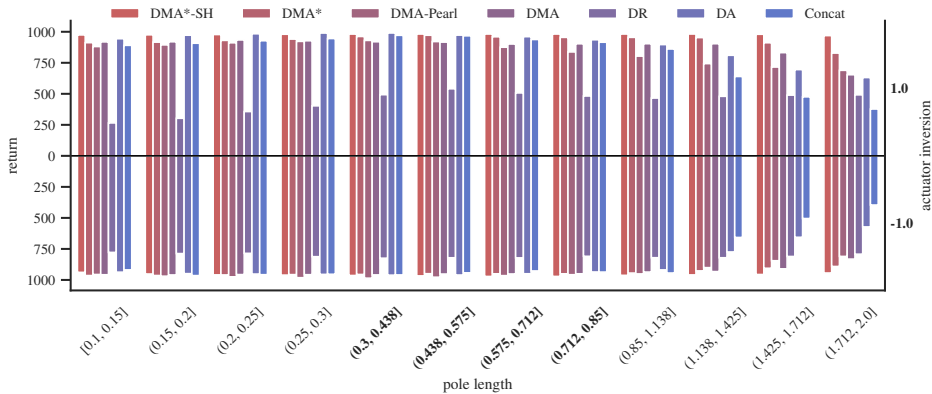
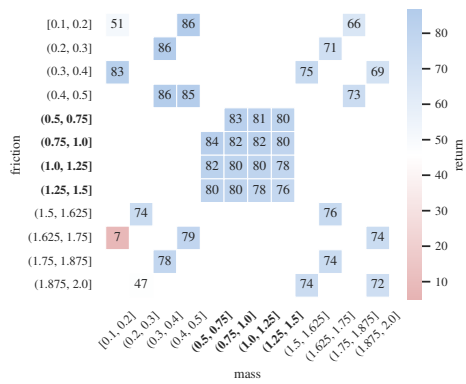
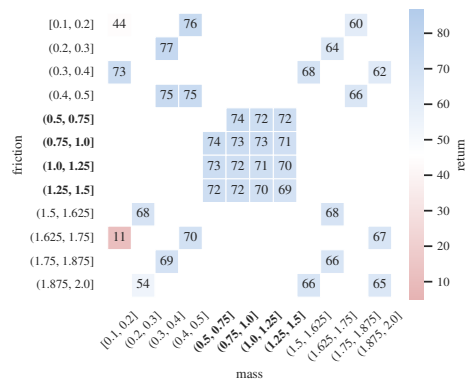


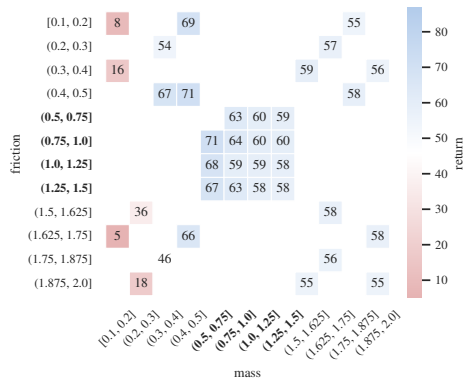
Figure 21: Bar plot for Cartpole to visualize AER for individual context instances and different methods. Bold labels refer to contexts used during training.



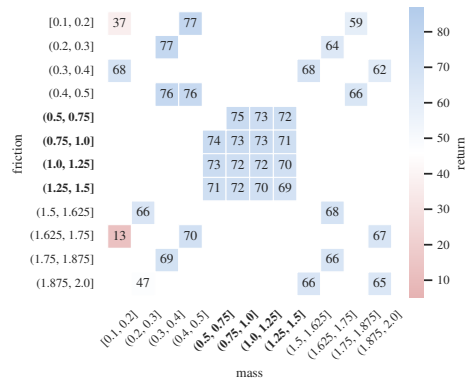
(a) DMA*-SH.



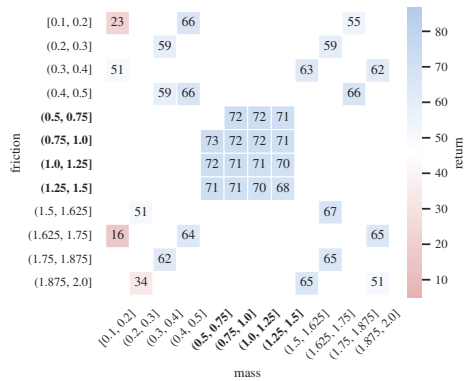
(b) DMA*.



(c) DMA.



(d) DR.

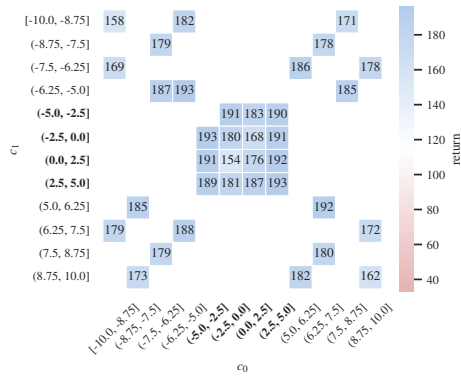


(e) Concat.

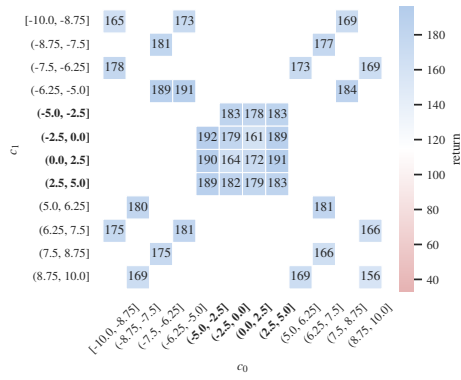


(f) DA.

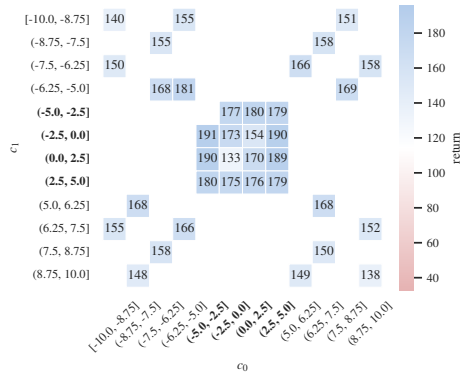
Figure 23: Heatmaps for DI-Friction to visualize AER for individual context instances. Bold labels refer to contexts used during training.



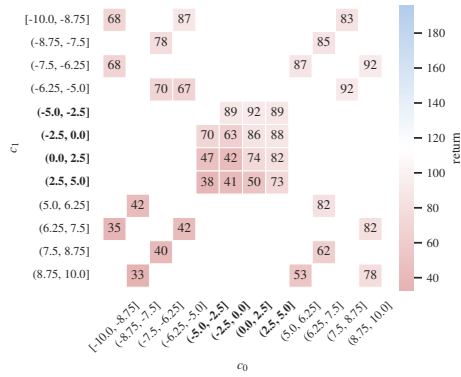
(a) DMA*-SH.



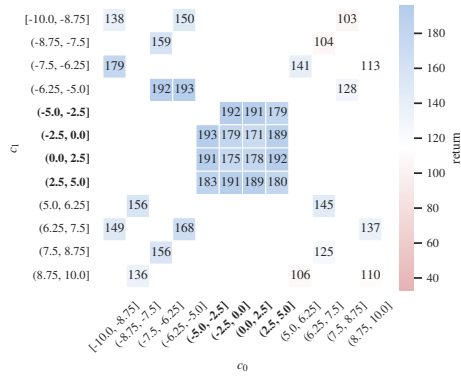
(b) DMA*.



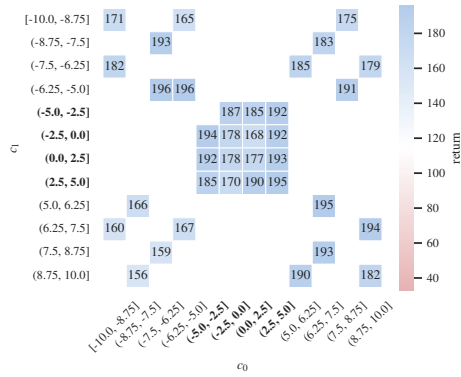
(c) DMA.



(d) DR.

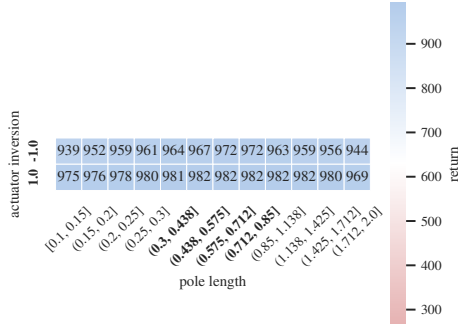


(e) Concat.

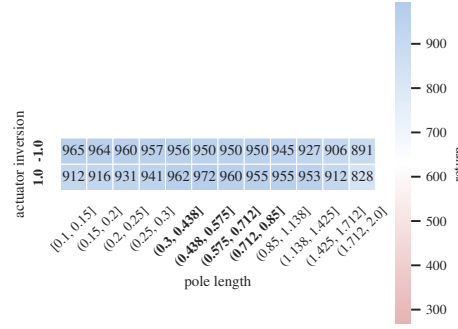


(f) DA.

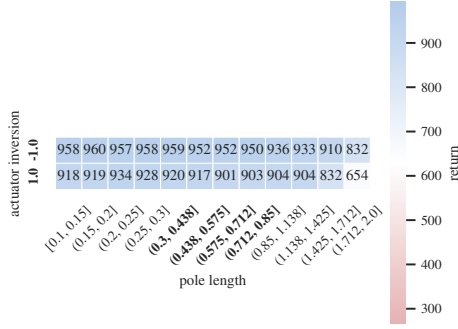
Figure 24: Heatmaps for ODE to visualize AER for individual context instances. Bold labels refer to contexts used during training.



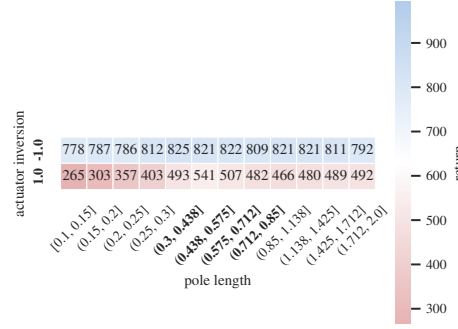
(a) DMA*-SH.



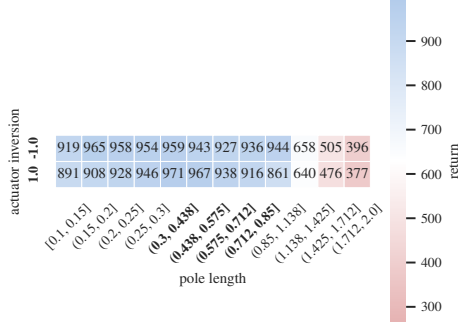
(b) DMA*.



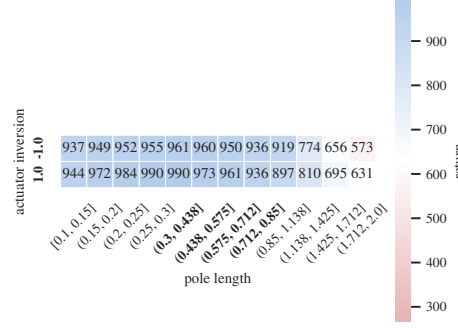
(c) DMA.



(d) DR.

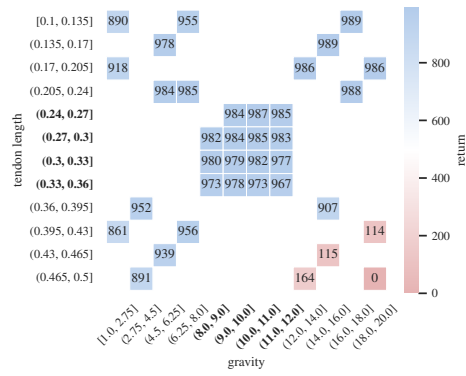


(e) Concat.

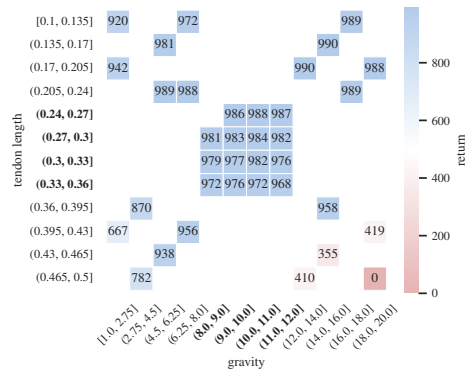


(f) DA.

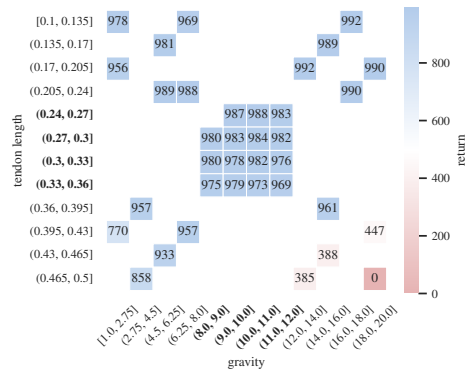
Figure 25: Heatmaps for Cartpole to visualize AER for individual context instances. Bold labels refer to contexts used during training.



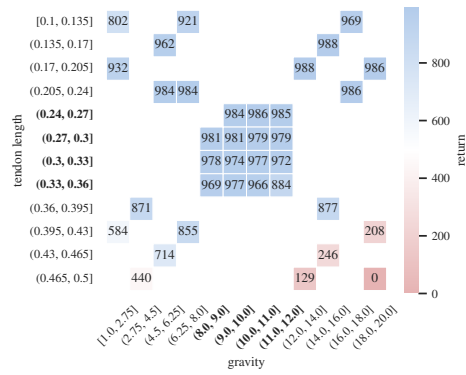
(a) DMA*-SH.



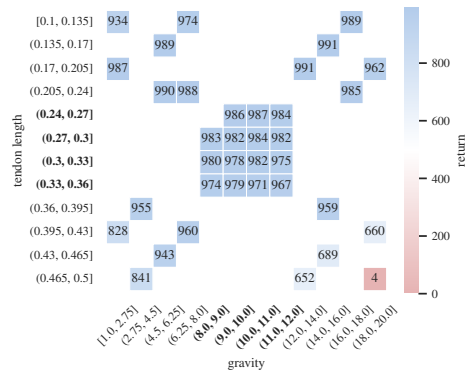
(b) DMA*.



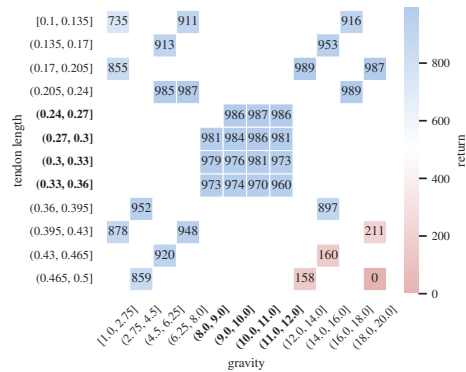
(c) DMA.



(d) DR.

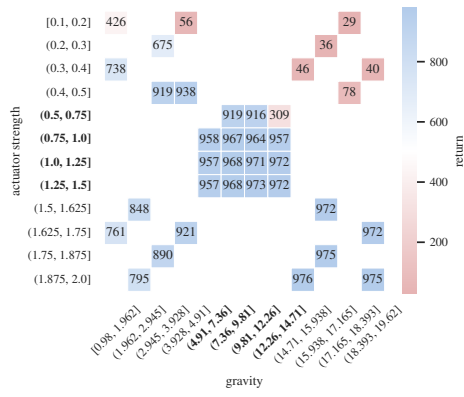


(e) Concat.

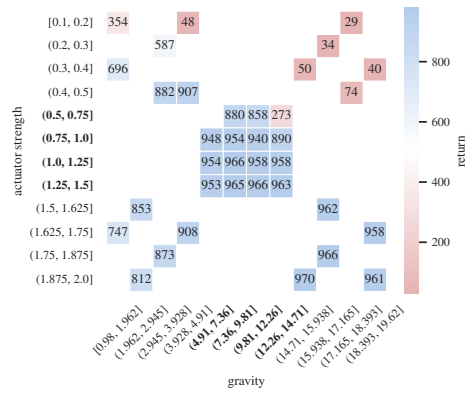


(f) DA.

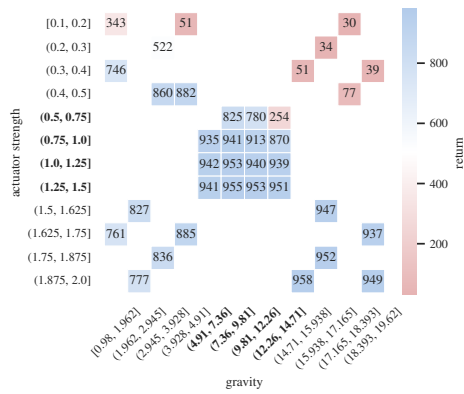
Figure 26: Heatmaps for BallInCup to visualize AER for individual context instances. Bold labels refer to contexts used during training.



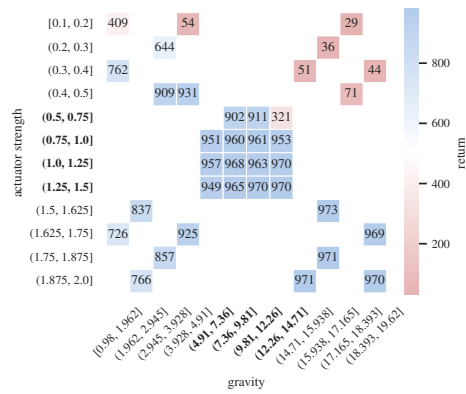
(a) DMA*-SH.



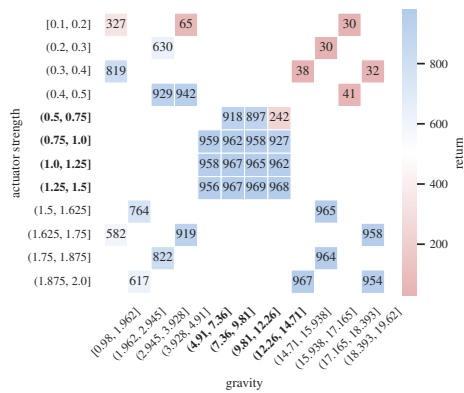
(b) DMA*.



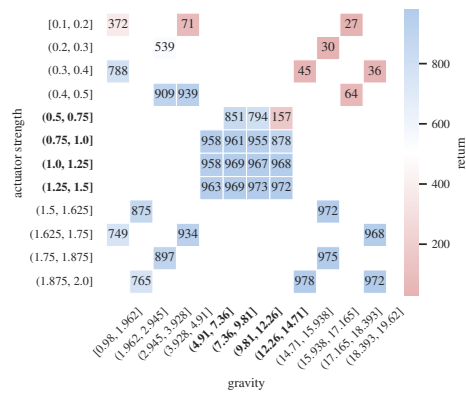
(c) DMA.



(d) DR.



(e) Concat.



(f) DA.

Figure 27: Heatmaps for Walker to visualize AER for individual context instances. Bold labels refer to contexts used during training.

G.3 Scalability in the Explicit Context Dimension

The standard ODE task (ODE-2) is governed by a differential equation parameterized by two context variables c_1 and c_2 (Table 5). To evaluate scalability with respect to context dimensionality, we extend this to ODE- k using higher-order polynomials (Appendix E), increasing the number of context parameters from 1 to 6. Figure 28 shows performance aggregated across these six variants. The results demonstrate that DMA*-SH scales favorably to higher context dimensions, whereas the context-aware Concat baseline struggles in both Eval-in and Eval-out regimes.

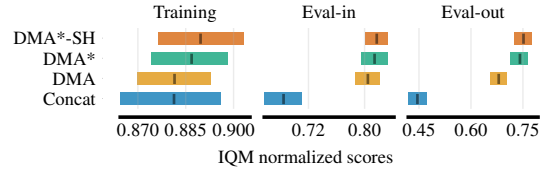


Figure 28: Interquartile mean (IQM) aggregated over six ODE variants, ODE-1, ODE-2, ... , ODE-6 to test for scalability with respect to context dimensionality.