TRUST, BUT VERIFY: TEST-TIME TASK INFERENCE IN OFFLINE META-REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Context-based offline meta-reinforcement learning adapts from a short history of transitions: an encoder turns the history into a task code, and a policy acts conditioned on that code. Outside the training range, return can fall for several reasons: the policy may be unable to solve the new task, the inferred code may be wrong, or the task may lie too far from anything seen in training. We derive an exact four-term decomposition of this lost return and show how to estimate its terms from checkpoint rollouts and data-collecting experts. The measurements reveal two failure modes. On manipulation tasks the frozen policy already solves the new task when given a suitable training code, so the choice of code fails; on locomotion tasks with shifted rewards the oracle-context code is already near the best available and leaves little code-selection headroom, while self-collected context can lose further return. Either way, choosing a code at test time is unreliable: the encoder can infer one from exploration episodes, and a decoder can select the training code that best explains them, but each wins where the other loses and no pre-deployment statistic reliably identifies which one to trust. We propose VeriCode, which explores with a spread of training codes and immediately keeps one seen to succeed. If none succeeds, it compares an encoder candidate and a likelihood candidate through one-episode verification, except when both heads identify the same training code; without a success signal, it compares candidates by return. Under independent evaluation episodes, the probability of selecting the worse of two fixed candidates decays exponentially with the verification budget. Across six Meta-World and six MuJoCo families and three algorithms, VeriCode is at or above the strongest comparator in every cell, raising Meta-World success from 0.52 to 0.83 and MuJoCo return from 200 to 267, at mean adaptation budgets of 5.5 and 9.4 episodes, respectively.

1 INTRODUCTION

The context-based recipe for offline meta-reinforcement learning (Rakelly et al., 2019; Li et al., 2021; Gao et al., 2023; Li et al., 2024) learns an encoder that turns a context of transitions into a task code and a policy that acts given the code; at test time the agent collects a context in the new task, infers a code, and acts. On OOD tasks—novel deployment situations in the broad sense of Kirk et al. (2023), instantiated here by parameters outside the training-task range—the return drops, and the benchmark tables that report the drop do not say why: the policy may be unable to solve the new task, the encoder may be handing it a bad code, or the self-collected context may be uninformative, and the three call for different remedies.

The usual response is to improve the representation and to read the improvement off the OOD table (Yuan & Lu, 2022; Gao et al., 2023; Li et al., 2024; Jia et al., 2024; Nakhaeinezhadfard et al., 2025). We instead ask the prior question: for a trained checkpoint and an OOD task, where does the lost return go? For any OOD task, any inferred code and any training task, the OOD gap is exactly the sum of four terms: the change in achievable return between the two tasks, the offline learner's own gap on the training task, the change in the training policy's return when it is carried over, and the difference between acting with the training code and with the inferred code. The first and third are bounded by a task metric and the fourth by how far the inferred code moved the actions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

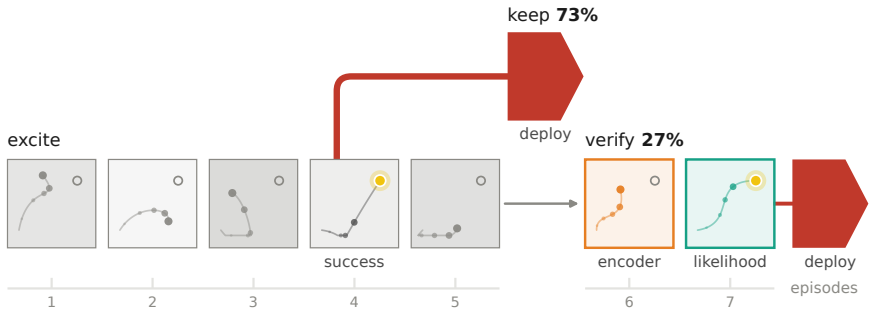


Figure 1: **Reuse outcomes before predicting.** Each tile is one episode in the same OOD task, drawn as the path of the hand with later steps darker; the ring is the goal, filled when it is reached. Five exploration episodes with a spread of training codes are already measurements: in 73 percent of trials one succeeds and its code is deployed with no further episode. Otherwise two inferred candidates are normally evaluated and compared; in the 10 percent of all trials where both heads identify the same training code, the implementation commits that agreement without an extra rollout.

The two benchmarks turn out to fail for different reasons. On cheetah-vel, extrapolation dominates. On ant-goal the best-training-code ceiling is already near the oracle-context encoder, while much of the recoverable online loss is contextual; both families therefore leave little code-selection headroom. On mass and friction shifts, by contrast, the best training code is 23 to 140 return above the encoder's. On Meta-World the frozen policy already solves 0.88 to 1.00 of the OOD tasks when given the best training code, and what fails is the code: the encoder reaches 0.54 with oracle context and the framework's online channel 0.25 (regime C).

This points at procedures that change nothing about training and act only on which code is deployed. Two such heads exist: the frozen encoder can re-infer the code from exploration episodes accepted as in-distribution by a return rule, which on Meta-World means episodes that contain a success, the principle of IDAQ (Wang et al., 2023); or a task-conditioned decoder can score every training code by the likelihood of the collected context and select the winner. The latter targets the population likelihood residual, which controls task distance only under the decoder and coverage assumptions stated below. Compared on the same exploration episodes and gated context over 6,480 paired trials, both heads round to 0.44; the likelihood head is ahead in four cells and behind in three. The obvious repair, switching between them on a confidence statistic, does not work either: margins, posterior mass, agreement across episodes and success counts are all weakly predictive of which head is right, and the best threshold, tuned per cell on the test tasks themselves, reaches 0.52 where a per-trial oracle over the two heads reaches 0.62.

VeriCode, for verified task code, replaces prediction by measurement whenever evidence is available (Figure 1). It explores with a spread of training codes and keeps any code observed to succeed. Only if none succeeds does it form encoder and likelihood candidates; it commits an agreed discrete code directly and otherwise tests the candidates, preferring the encoder on ties. Under the stated tail, decoder and coverage assumptions, likelihood selection identifies the population-likelihood-best training code at an exponential rate for idealised independent context samples and its residual controls task distance. On six Meta-World families and three algorithms, VeriCode reaches 0.83 success at a mean budget of 5.5 episodes, above the best competitor by more than two standard errors in all eighteen cells. The alternatives reach 0.42 and 0.45 for IDAQ at five and seven episodes, 0.52 for the strongest hybrid, and 0.25 for the framework channel; keeping an observed success alone reaches 0.73 and candidate comparison adds about 0.10. On six MuJoCo families without a success signal, return comparison is at or above the best competitor in all eighteen cells and significantly above in thirteen. Across all thirty-six cells, it is significantly below the best competitor in none.

Our decomposition is not a new type of bound: under shared dynamics and linear rewards, its source-task telescoping structure is analogous to the successor-feature transfer bound of Barreto et al. (2017), while distance-to-a-seen-task and context-inference terms appear in Sodhani et al. (2022), Zhang et al. (2021) and Fu et al. (2023). What we add is twofold. Offline, we add the target quantity and its split: the return of a deployed offline meta-RL policy on an OOD task, decomposed

with an explicit offline term and with the task metric kept separate from learned code displacement; the corresponding expert-reference terms are estimated on trained checkpoints. Online, we add the excitation condition, the comparison of the two heads, and a verification rule that does not require a calibrated confidence threshold.

2 Related work

Context-based offline meta-RL and online adaptation. The lineage runs from PEARL (Rakelly et al., 2019) through FOCAL (Li et al., 2021), CORRO (Yuan & Lu, 2022), CSRO and its non-prior collection rule (Gao et al., 2023), UNICORN (Li et al., 2024) and ER-TRL (Nakhaeinezhadfard et al., 2025), with work on data limitations (Zhou et al., 2024; Jia et al., 2024) and on cross-task context (He et al., 2026); we evaluate CSRO, UNICORN and ER-TRL without changing their training. Training-side approaches to the same failure (Mitchell et al., 2021; Zintgraf et al., 2020; Liu et al., 2021; Pong et al., 2022; Ma et al., 2025) are discussed in Appendix L. Zhang et al. (2025a) bound the return loss on the training tasks by a representation shift under the policy-side Lipschitz condition our last term uses. At test time, MIER (Mendonca et al., 2020) identifies a task model and relabels experience under it; IDAQ (Wang et al., 2023) samples task hypotheses from its current belief, rolls out the corresponding conditioned policy, and re-infers from episodes accepted by a return-based in-distribution filter; and Meta-DT (Wang et al., 2024) collects its context with self-guided prompts. VeriCode differs from IDAQ in what it trusts: IDAQ accepts an episode and then trusts the encoder to read a code out of it, whereas we reuse measured outcomes and normally test unmeasured candidates directly. Test-time selection also appears in Liu & Zhang (2026) and, for behaviour foundation models, in Rupf et al. (2026), and outside meta-RL in active offline policy selection (Konyushkova et al., 2021); Appendix L states the axis of difference. Widening the task distribution at training time (Ajay et al., 2022; Kim et al., 2025) is complementary.

Metric-based transfer bounds and task models. Bisimulation metrics compare states through rewards and transition laws (Ferns et al., 2004), and related task-level constructions control transfer across MDPs (Barreto et al., 2017; Sodhani et al., 2022; Zhang et al., 2021; Fu et al., 2023; Lecarpentier et al., 2021; Sabbioni et al., 2023; Zhang et al., 2025b; Cho et al., 2026). Work on latent MDPs gives complementary identification and regret guarantees (Kwon et al., 2021). The KL bounds of Liu (2025) require the relevant absolute-continuity conditions, whereas metric bounds need not. Model-based approaches (Lin et al., 2022; Nakhaei et al., 2026; Qian et al., 2026) learn richer task models than the decoder we use only to score training codes, while FLORA models task-conditioned feature distributions with a normalising flow (Wang et al., 2026). VeriCode acts after backbone training and can therefore wrap such approaches rather than replace them.

3 Anatomy of the out-of-distribution gap

In this section, we make precise the return a deployed context-conditioned policy leaves on the table in an OOD task, and write it as an exact identity whose terms are bounded by a task metric, an offline gap, and a code displacement. We begin with notation and the metric, then the lemmas, the theorem, and what each term measures.

We study a family of Markov decision processes $\{M_\theta = (\mathcal{S}, \mathcal{A}, P_\theta, r_\theta, \gamma, \rho)\}_{\theta\in\Theta}$ that share the state space, action space, discount $\gamma \in [0, 1)$, and initial distribution ρ, and differ in the transition kernel $P_\theta(\cdot \mid s, a)$ and reward $r_\theta(s, a, s') \in [r_{\min}, r_{\max}]$, where $0 \in [r_{\min}, r_{\max}]$. We write $\Delta_r := r_{\max} - r_{\min}$, and model termination by an absorbing state with reward zero.

Offline meta-training sees n training tasks $\theta_1, \dots, \theta_n$ through datasets $D_1, \dots, D_n$ and returns an encoder together with a code-conditioned policy $\pi(a \mid s, z)$. For each training task we fix one *training code* $\bar{z}_i$, the encoder's output averaged over contexts drawn from D_i. We write π_z for the stochastic policy with code frozen at z and $\mu_z(s)$ for its deterministic action. To keep the two evaluation modes distinct, let $\pi_z^{\mathrm{ev}} := \pi_z$ under stochastic evaluation and $\pi_z^{\mathrm{ev}} := \delta_{\mu_z}$ under deterministic evaluation; the experiments use the latter. At test time we face an *OOD task* θ^*, obtain a context through some channel, infer a code $\hat{z}$, and act with $\pi_{\hat{z}}^{\mathrm{ev}}$. The return of a policy q in task θ is $J_\theta(q) := \mathbb{E}^{q,\theta} \sum_{t\geq 0} \gamma^t r_\theta(s_t, a_t, s_{t+1})$ and $J^*(\theta) := \sup_q J_\theta(q)$. The quantity we want to understand is the *OOD gap* $J^*(\theta^*) - J_{\theta^*}(\pi_{\hat{z}}^{\mathrm{ev}})$. The theory holds for any target task; "OOD"

describes the experiments. Suprema over state-action pairs are over a reachable set $\mathcal{K}$ that contains every occupancy support (Appendix A.1).

Task pseudometric. We measure how different two tasks are by

$$d(\theta,\theta') := \sup_{(s,a)\in\mathcal{K}} \sup_{s'} |r_\theta(s,a,s') - r_{\theta'}(s,a,s')| + \frac{\Delta_r}{1-\gamma} \sup_{(s,a)\in\mathcal{K}} \mathrm{TV}\big(P_\theta(\cdot \mid s,a), P_{\theta'}(\cdot \mid s,a)\big). \tag{1}$$

The first term is the most the two tasks can disagree about the reward of a transition; the second is the most they can disagree about where it leads, weighted by the largest span of future values. Distinct task parameters can induce the same model on $\mathcal{K}$, so d is formally a pseudometric (and a metric on the corresponding equivalence classes). For reward-shift families the transition term vanishes, and parameter displacement gives an analytic upper bound on the reward term ($|v-v'|$ for target velocities and $\|g-g'\|_1$ for an ℓ_1 goal reward), with equality when the reachable set contains an appropriate one-sided state. For the shaped Meta-World rewards, goal displacement is only an empirical proxy. For deterministic MuJoCo dynamics shifts, Appendix A.3 gives a conditional Wasserstein form d_{W} whose transition term uses a common coefficient that dominates the required value-Lipschitz constants.

Lemma 1 (Values are Lipschitz in the task metric)**.** *For every policy π, history-dependent policies included, $|J_\theta(\pi) - J_{\theta'}(\pi)| \le d(\theta,\theta')/(1-\gamma)$ and $|J^*(\theta) - J^*(\theta')| \le d(\theta,\theta')/(1-\gamma)$.*

That is, running the same policy in two tasks that are d apart changes the return by at most $d/(1-\gamma)$, and the optimal returns are at least as close. This is the classical simulation lemma with a task metric in place of a model error (Lecarpentier et al., 2021; Sodhani et al., 2022), and a code-conditioned policy is covered only if the code is the same frozen vector in both tasks, which is how the harness runs it (Appendix A).

The second ingredient is a performance difference bound within one task (Lemma 2, Appendix A). For stochastic policies, the return difference is at most $\frac{\Delta_r}{(1-\gamma)^2}$ times the expected total variation between them under the discounted occupancy (Kakade & Langford, 2002). For deterministic policies μ, μ' with an L_Q-Lipschitz critic it is at most $\frac{L_Q}{1-\gamma}\,\mathbb{E}_{s\sim d_\theta^\mu}\|\mu(s)-\mu'(s)\|_2$, the form that matters because evaluation runs $\mu_z(s) = \tanh(m(s,z))$.

Three quantities describe the learned system (Appendix A). The first is the theoretical *offline gap* $\varepsilon_i := J^*(\theta_i) - J_{\theta_i}(\pi^{\mathrm{ev}}_{\bar z_i}) \ge 0$ on training task i. Because our backbones are behaviour-regularised (Wu et al., 2019) rather than conservative (Kumar et al., 2020) and carry no pessimism guarantee (Jin et al., 2021), the experiments replace J^* with the data-collecting expert and measure the resulting reference gap. The second is the *policy sensitivity* L_μ in $\sup_s\|\mu(s,\bar z_i) - \mu(s,\hat z)\|_2 \le L_\mu\|\hat z - \bar z_i\|_2$ for the single pair compared (L_π in total variation for stochastic evaluation). The third, needed only for deterministic evaluation, is the *action-Lipschitz constant* L_Q of $Q_{\theta^*}^{\mu_{\bar z_i}}$ on states visited by $\mu_{\hat z}$. The sensitivity and critic probes are sampled diagnostics, not exact global constants.

Theorem 1 (OOD decomposition)**.** *For any target task θ^*, any code $\hat z$, and any training index i with fixed code $\bar z_i$, the exact identity*

$$\begin{aligned} J^*(\theta^*) - J_{\theta^*}(\pi^{\mathrm{ev}}_{\hat z}) = {} & \underbrace{\big[J^*(\theta^*) - J^*(\theta_i)\big]}_{A_i} + \underbrace{\big[J^*(\theta_i) - J_{\theta_i}(\pi^{\mathrm{ev}}_{\bar z_i})\big]}_{\varepsilon_i} \\ & + \underbrace{\big[J_{\theta_i}(\pi^{\mathrm{ev}}_{\bar z_i}) - J_{\theta^*}(\pi^{\mathrm{ev}}_{\bar z_i})\big]}_{B_i} + \underbrace{\big[J_{\theta^*}(\pi^{\mathrm{ev}}_{\bar z_i}) - J_{\theta^*}(\pi^{\mathrm{ev}}_{\hat z})\big]}_{C_i(\hat z)} \end{aligned} \tag{2}$$

holds, with $|A_i|, |B_i| \le d(\theta^*,\theta_i)/(1-\gamma)$. For a dynamics shift, the same bounds hold with d_{W} under Lemma 1-W, or for Dirac kernels with the pointwise realised-reward pseudometric $d_{\mathrm{W}}^{\mathrm{pt}}$ in equation 6. In either case the relevant common-coefficient conditions must hold both for $\pi^{\mathrm{ev}}_{\bar z_i}$ and for the near-optimal policies used to compare the two optimal returns. If the policy-sensitivity assumption holds, then the stochastic branch satisfies the first bound below; if the action-Lipschitz assumption also holds, the deterministic branch satisfies the second:

$$\begin{aligned} & C_i(\hat z) \le \tfrac{\Delta_r L_\pi}{(1-\gamma)^2}\|\hat z - \bar z_i\|_2 \quad \textit{(stochastic evaluation)}, \\ & C_i(\hat z) \le \tfrac{L_Q}{1-\gamma}\,\mathbb{E}_{s\sim d_{\theta^*}^{\mu_{\hat z}}}\|\mu(s,\hat z) - \mu(s,\bar z_i)\|_2 \le \tfrac{L_Q L_\mu}{1-\gamma}\|\hat z - \bar z_i\|_2 \quad \textit{(deterministic evaluation)}. \end{aligned} \tag{3}$$

Writing $\bar{C}_i(\hat{z})$ for the applicable right-hand side, $J^*(\theta^*) - J_{\theta^*}(\pi^{\mathrm{ev}}_{\hat{z}}) \le \min_i\{2d(\theta^*, \theta_i)/(1-\gamma) + \varepsilon_i + \bar{C}_i(\hat{z})\}$; under the preceding Wasserstein conditions, d may be replaced by d_{W}, or by $d^{\mathrm{pt}}_{\mathrm{W}}$ in the deterministic realised-reward setting, for dynamics shifts.

The identity splits the loss on a new task into four algebraic terms. The first and third brackets measure the effect of changing from training task i to the target while holding the relevant policy fixed; together they form the *extrapolation term*. The second is the *ID gap* for the chosen source task and code, which cannot be changed while that pair is held fixed. The last is the *latent term*: the return lost, or gained, by using the inferred code instead of the training code. Under the stated regularity conditions, it is bounded by how far the inferred code moves the actions.

Proof idea. The identity is a telescoping with no assumption; Lemma 1 bounds A_i and B_i and Lemma 2 bounds C_i (Appendix A).

The intuition comes from measuring the brackets. We never observe J^*. On MuJoCo the experiments replace it by the return $E(\theta)$ of the data-collecting expert; the algebraic identity survives, while a bound requires the expert cross-task regret of Appendix A.6. For Meta-World success, $E = 1$ is instead a common reference ceiling, used only for the descriptive decomposition because Corollary 1 rules out a generally non-trivial success-rate bound. Writing N, O and L for the returns of the parameter-nearest training code, of the encoder's code inferred from the OOD task's own offline data and of the framework's online channel, the reference identity reads $E - L = \Delta_{\mathrm{extrap}} + \varepsilon + \Delta_{\mathrm{latent}} + \Delta_{\mathrm{ctx}}$ with $\Delta_{\mathrm{extrap}} := (E - N) - \varepsilon$, $\Delta_{\mathrm{latent}} := N - O$, negative where the encoder's own code beats the nearest training code, and $\Delta_{\mathrm{ctx}} := O - L$ the price of a self-collected rather than an oracle context. Returns are undiscounted over $H = 200$ steps (Appendix A.7); for Meta-World we bound only the dense return and report success descriptively.

4 Diagnosis: where the return is lost

Protocol. We evaluate frozen checkpoints of CSRO (Gao et al., 2023), ER-TRL (Nakhaeinezhadfard et al., 2025) and UNICORN (Li et al., 2024), trained with each method's own code on twenty training tasks, on four MuJoCo families (Todorov et al., 2012) and six Meta-World families (Yu et al., 2020); the OOD split is the band of ten tasks just outside the training range (Appendix K). A further Meta-World band is not used in the main diagnosis or method comparison because the best training code itself reaches only 0.00 to 0.08 there on five of the six families; it appears only in the earlier metric diagnostic of Appendix J. Walker-mass and walker-friction are excluded from this diagnosis under a rule fixed before the method was designed: the ID gap exceeds half of $E - L$ for at least two algorithms (Appendix E). They return in Section 6. Each OOD task is rolled out under five channels: E, O, L, N and all twenty training codes, which give the cross-validated best code B_{cv}.

Two regimes. Table 15 and Figure 6 in the appendix give the four measured terms per environment and algorithm. Extrapolation is largest in eight of twelve MuJoCo cells—all six hopper cells and two cheetah cells—whereas ant-goal is context- or ID-gap-dominated; the nearest-code latent term is near zero or negative throughout. Code-selection headroom is within noise on cheetah-vel and ant-goal but 23 to 78 return on hopper. On Meta-World the picture inverts: in reach, reach-wall and pick-place the context term dominates in eight of nine cells, the ID gap is zero in five of those nine cells, the encoder with oracle context reaches 0.36 to 0.76 and the framework's online channel 0.00 to 0.33. This is *regime C*, context dominated: the policy is not the problem; the self-collected context drives the wrong code. Push, soccer and plate-slide sit in between.

The ceiling. The B_{cv} column motivates the method: on every Meta-World family the best of the twenty training codes solves 0.88 to 1.00 of the OOD tasks, against 0.19 to 0.84 for the encoder's code with oracle context, so the frozen policy already solves the tasks and only needs the right code. Any rule that picks a training code is bounded by B_{cv}, so $B_{\mathrm{cv}} - O$ is the most it can gain over the encoder's oracle-context code: 0.1 to 0.8 on Meta-World, within noise on cheetah-vel and ant-goal, 23 to 140 return on hopper and walker. Online the encoder does not get an oracle context: on Meta-World it collapses from 0.54 to 0.25, so even the reward-shift families leave a context term to recover.

5 VERICODE: REUSE OUTCOMES AND VERIFY CANDIDATES

The procedure. VeriCode has three steps. *Excitation*: on Meta-World it runs $E = 5$ exploration episodes in the OOD task, each with a different training code chosen by farthest-point sampling in code space. *Keep what worked*: if any exploration episode succeeds, the agent deploys the successful training code with the highest return and stops, spending no episode beyond excitation. *Compare the heads*: only when excitation produces no success does the full context feed two post-training heads. The frozen encoder re-infers $\hat{z}_{\text{enc}}$. Likelihood selection scores every training code by the log-likelihood of the context under a task-conditioned decoder $p_\psi(x \mid s, a, z)$, with $x = (\Delta s, r)$, and returns $\bar{z}_{\hat{\imath}}$. The decoder is homoscedastic and is trained once, without changing the policy or encoder, on every snapshot tier of the training buffers with $z = \bar{z}_i$; Appendix L ablates these design choices. If $\bar{z}_{\hat{\imath}}$ is the training code nearest $\hat{z}_{\text{enc}}$, the rule commits that agreed discrete choice without another rollout. Otherwise it runs one episode with $\hat{z}_{\text{enc}}$ and, unless $\bar{z}_{\hat{\imath}}$ was already observed during excitation, one with $\bar{z}_{\hat{\imath}}$. It commits to a successful candidate, uses comparison return if both fail, and breaks exact ties toward $\hat{z}_{\text{enc}}$. Thus excitation supplies direct outcomes and may supply informative context; Theorem 2-B characterises the required visitation condition. Selection controls a choice among training tasks under its stated assumptions (Theorem 2-A), and fresh independent verification controls the risk of choosing between fixed candidates (Proposition 3). The agreement shortcut is the one exception to direct verification and occurs in 10 percent of trials (Appendix D). The values of E, the verification budget, and the tie-break were fixed before the main experiment; the MuJoCo variant is specified below and in Appendix K.

Selection as least squares. With a fixed width, S_i is minus half the sum of squared residuals of code i, so likelihood selection is least-squares selection of the training model that best predicts the context; its population counterpart $R_i := \mathbb{E}_{\nu^*}\|(\bar{x}^* - f_i)/\sigma\|^2$ defines a fixed-tie likelihood-best code $i_{\text{L}} \in \arg\min_i R_i$ and the *excitation gap* $\Delta_i := \frac{1}{2}(R_i - R_{i_{\text{L}}})$ that exploration has to make positive. The theorem uses three explicit quantities: a global decoder-error bound δ, a sub-Gaussian parameter σ_ℓ^2, and a coverage pair $(C_{\mathcal{K}}, \kappa)$. The experiments provide diagnostics related to them, not measurements of the global constants (Appendix B).

Theorem 2-A (Finite-sample bound for the likelihood-best training code)**.** *Assume a deterministic simulator, the setting of Theorem 1, and Assumptions (D0) to (D3) of Appendix B. For dynamics shifts, also fix one coefficient c_P satisfying the common-coefficient conditions of the pointwise realised-reward Wasserstein form equation 6 for every training-code evaluation policy and for the near-optimal policies in both comparison directions. Let $\Delta_{\min} := \min_{i \neq i_{\text{L}}} \Delta_i$. With probability at least $1 - (n-1)\exp\big(-m\Delta_{\min}^2/(2\sigma_\ell^2)\big)$ over the idealised i.i.d. context, the selected code is the likelihood-best code, $\hat{\imath} = i_{\text{L}}$, and*

$$J^*(\theta^*) - J_{\theta^*}(\pi^{\text{ev}}_{\bar{z}_{\hat{\imath}}}) \;\le\; \frac{2\big(C_{\mathcal{K}}\, c_\sigma \sqrt{\min_i R_i} + \kappa + \delta\big)}{1-\gamma} + \varepsilon_{i_{\text{L}}}, \qquad c_\sigma := \sigma_r + c_P\, \sigma^{\Delta s}_{\max}, \tag{4}$$

with c_P the transition coefficient of the metric and σ_r, $\sigma^{\Delta s}_{\max}$ the decoder widths on the reward and the largest next-state coordinate. In the shared-dynamics reward-only specialization, take $c_P = 0$ and omit the next-state coordinates from R_i and the scores.

The theorem controls the code that minimises the population context residual R_i. Under decoder and coverage assumptions, that residual controls the selected code's true task distance. It need not minimise the task pseudometric itself, the source-task offline gap ε_i, or the full bound in Theorem 1. When $\Delta_{\min} > 0$, the finite-sample error term decays exponentially in the idealised number of independent context transitions times the squared excitation gap; actual trajectory contexts require additional dependence assumptions. Since the selected code is a training code, its gain over the encoder is at most the true code-set ceiling gap $B - O$, where $B := \max_i J_{\theta^*}(\pi^{\text{ev}}_{\bar{z}_i})$; this gap is estimated by $B_{\text{cv}} - O$. Figure 5 reports softmax mass as an in-distribution score-concentration diagnostic, not an OOD correctness certificate. The proof and finite-horizon form are in Appendix B.

Having established what likelihood selection controls when its assumptions hold, the remaining question is how to make the context informative online. This is the excitation condition, and it is where the two families of rewards part ways.

Theorem 2-B (Excitation: selection needs an informative exploration distribution)**.** *Let ν_{e} be the state-action visitation of an exploration policy on θ^* and let the context be m transitions from $\nu_{\text{e}} \otimes$*

P_{θ^*}. *(i)* Sufficiency. *Under all assumptions of Theorem 2-A, with* $\nu^* = \nu_e$*, if* $\Delta_{\min} > 0$*, which we write* $\Delta_{\min}(\nu_e) > 0$*, then the selection-probability and return conclusions of that theorem hold. (ii)* Impossibility under observational indistinguishability. *Fix* m *and suppose two tasks* θ_1^*, θ_2^* *induce the same law of the length-*m *context under the exploration procedure. Suppose also that their unique return-best training codes* $i_k := \arg\max_{i\in[n]} J_{\theta_k^*}(\pi_{\tilde{z}_i}^{\mathrm{ev}})$ *satisfy* $i_1 \neq i_2$*. Then every selection rule, randomised or not, satisfies* $\max_{k\in\{1,2\}} \Pr_{\theta_k^*}(\hat{\imath} \neq i_k) \geq \frac{1}{2}$.

Part (i) says that, under the stated tail, decoder, coverage, and transfer conditions, a positive but small gap can be offset by a longer idealised i.i.d. context; when the gap is zero, this concentration argument is inconclusive. Part (ii) is a separate impossibility result: if the available length-m observation law is identical for tasks requiring different choices, no selector can distinguish them at that context length. This is the code-space form of an uninformative experiment (Ljung, 1999; Narendra & Annaswamy, 1987), and the online counterpart of the offline ambiguity of Dorfman et al. (2021). For goal rewards $r = \phi(\|h(s) - g\|)$, an exploration that barely moves the hand can leave pairs of goals observationally indistinguishable. Appendix C proves two quantitative cases: bracketing excitation for quadratic goal rewards, for which $K \geq d_{\mathrm{goal}} + 1$ episode means that affinely span $\mathbb{R}^{d_{\mathrm{goal}}}$ are a sufficient between-episode construction; and monotone excitation for one-dimensional symmetric rewards, where exploring at one intermediate speed leaves the mirror speed indistinguishable.

The theorem explains a pattern rather than certifying a particular method. IDAQ's diverse hypothesis-conditioned exploration is aligned with informative excitation, but it satisfies the condition only when the induced visitation has $\Delta_{\min}(\nu_e) > 0$. Its acceptance rule encourages in-support encoder contexts, rather than proving $\kappa \approx 0$ or small decoder error, and it may accept no episode (Appendix C.5).

Verification. The operational principle is to reuse direct measurements whenever they are available: a training code that succeeded during excitation is committed at no extra cost, which is why the budget stays close to E, while otherwise the two inferred candidates are normally tested in the target task. Section 6 shows that neither head dominates and that none of the six logged context statistics reliably predicts which one is right; outcome comparison instead measures performance under the deployment distribution. The agreement shortcut described above is the explicit exception to a direct test. Without a success signal, on the dense-reward MuJoCo families, the rule compares candidates by return. Two changes are motivated by the excitation analysis: two of the five exploration episodes use the prior code $z = 0$, whose re-inferred code, the framework's own online channel, joins the candidates, and the likelihood head contributes its top three codes. The incumbent belongs in the candidate set for the same reason the encoder's code is preferred on ties: verification chooses only among what it is given, and on the symmetric cheetah-vel reward the prior code is the only one that explores on one side of the candidate speeds (Proposition B.2).

Proposition 3 (Verification risk for two fixed candidates)**.** *Let* z_1, z_2 *be two fixed candidate codes for* θ^* *whose evaluation episodes succeed independently with probabilities* $p_1 \geq p_2$*, and let the agent run* v *fresh verification episodes with each and commit to* z_2 *only if its episodes succeed more often than those of* z_1*. Then* $\Pr(\text{commit to } z_2) \leq \exp\big(-v(p_1 - p_2)^2\big)$ *whenever* $p_1 > p_2$*, and if success is a deterministic function of the code in* θ^**, the committed code attains* $\max\{p_1, p_2\}$ *with probability one for every* $v \geq 1$.

The proof is Hoeffding's inequality on the difference of two binomial rates (Appendix D). Its return version is Proposition 3-R in the same appendix: $\Pr(\text{commit to a code } \Delta \text{ below the best}) \leq \exp(-v\Delta^2/(2\Delta_G^2))$ for returns of range Δ_G, with an expected loss of at most $(K-1)\Delta_G/\sqrt{ev}$ over K fixed candidates. This is the simple regret of fixed-budget best-arm identification under uniform allocation (Audibert et al., 2010; Bubeck et al., 2011). A union bound gives $(K-1)\exp(-v\Delta_p^2)$ in the success-probability version. The proposition needs neither a coverage slack κ, a decoder error δ, nor a threshold calibrated on training tasks because it measures outcomes in the target task. It does, however, require candidates fixed independently of fresh verification outcomes. The guarantee applies directly to fresh pairwise verification, not to the full adaptive rule. Reusing a binary success outcome is exact in our seeded deterministic Meta-World harness; MuJoCo return reuse remains an empirical shortcut. More than 15,000 task-code pairs in the diagnosis have three-for-three agreement on success (Appendix D), so one episode identifies the success of a fixed code in this harness.

Table 1: **Meta-World OOD success.** Mean over ten tasks, six adaptation trials, three algorithms and six seeds (108 frozen checkpoints). *Keep* commits to a successful exploration code; *oracle* is the best VeriCode candidate per trial. The last row gives adaptation episodes; Table 13 gives per-cell errors.

family	framework	$IDAQ_5$	$IDAQ_7$	encoder	likelihood	hybrid	keep	**VeriCode**	oracle
reach	0.06	0.26	0.32	0.32	0.39	0.49	0.53	**0.72**	0.72
reach-wall	0.25	0.53	0.58	0.53	0.55	0.74	0.66	**0.88**	0.88
push	0.37	0.36	0.38	0.41	0.49	0.41	0.88	**0.92**	0.93
pick-place	0.12	0.40	0.44	0.42	0.30	0.47	0.66	**0.74**	0.74
soccer	0.22	0.28	0.29	0.29	0.39	0.28	0.82	**0.86**	0.86
plate-slide	0.49	0.69	0.70	0.65	0.56	0.74	0.86	**0.87**	0.87
mean	0.25	0.42	0.45	0.44	0.44	0.52	0.73	**0.83**	0.84
episodes	2	5	7	5	5	5	5.0	**5.5**	5.5

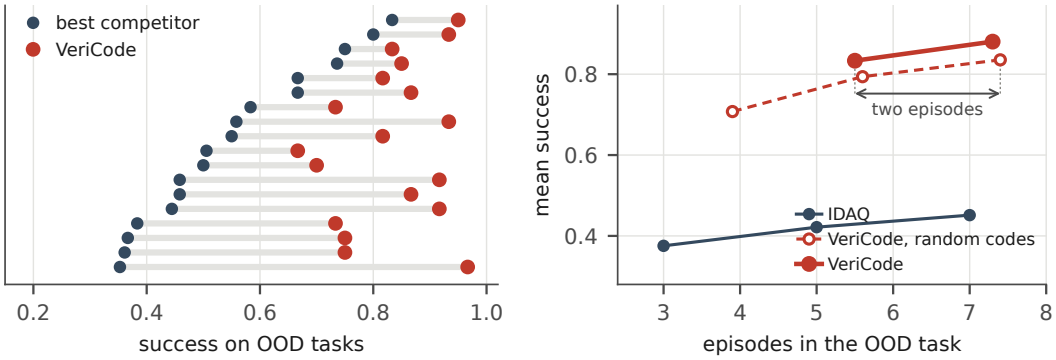


Figure 2: **Verification beats prediction (Meta-World).** Left: one row per cell of the grid, six families × three algorithms, showing that cell's best competitor and VeriCode, mean over six seeds; all 18 rows are above their best competitor by more than two standard errors. Right: success against the interaction budget. Rolling training codes out one at a time and keeping the first that works needs 8.1 episodes to reach what VeriCode reaches with 5.5. This is an exact reference from the all-codes rollouts. Both saturate at the same ceiling: the 0.96 of trials in which some training code works. The random-code variant is in Table 12.

6 Experiments

We compare VeriCode with the framework channel, IDAQ at two budgets, both heads and a hybrid that re-infers with the encoder when exploration yields an accepted successful episode and otherwise falls back to likelihood selection.

Main result. Table 1 reports test-time success. VeriCode reaches 0.83, against 0.25 for the framework's own online channel, 0.42 and 0.45 for IDAQ at five and seven episodes, 0.44 for either head on the same context and 0.52 for the strongest hybrid, and it spends fewer episodes than the seven-episode IDAQ it beats by 0.38 ± 0.02 paired over the 108 checkpoints. It is above the best competitor of the cell by more than two standard errors in all eighteen cells (Figure 2, left), and the gain is largest where the diagnosis predicted it: success roughly doubles on push and soccer.

Where the gain comes from. Excitation with five farthest-point training codes produces a successful episode in 73 percent of trials. Committing to the code that produced the success adds no verification episode and reaches 0.73, whereas IDAQ re-infers a code *from* its successful episodes and reaches 0.42. Candidate comparison adds about 0.10, lifting the mean to 0.83. A per-trial oracle over VeriCode's candidates reaches 0.835 against the rule's 0.833, so the remaining headroom is in the candidate set rather than the decision rule.

Against blind search. Every candidate is a training code, so the honest alternative is to spend the same episodes rolling codes out one at a time and keeping the first that works. The diagnosis rolled

out all twenty codes on every OOD task, so this reference is exact: blind search reaches 0.715 at five episodes and needs 8.1 to reach VeriCode's 0.833, both saturating at 0.963 (Figure 2). The two inferred heads buy the difference: they cost half an episode and add 0.10, where blind trials need three more for the same gain. They are verified rather than ranked because nothing computable ranks them: both round to 0.44 on the same context, a per-trial oracle over them reaches 0.62, and none of the six logged confidence statistics predicts which is right (Appendix L).

Ablations. Table 12 isolates the ingredients. Random rather than farthest-point exploration costs 0.040 ± 0.013 success on Meta-World and 12.7 ± 3.9 return on MuJoCo; within the random-code variant, reducing the exploration budget from $E = 5$ to $E = 3$ costs 0.083 ± 0.007. Removing the success gate leaves 0.79, while training the decoder only on the curated window costs the rule 0.044 ± 0.008. Dropping the keep step reaches 0.60 at 6.6 episodes. Under a stochastic actor, a kept code succeeds again in only 0.67 of trials and the rule falls to 0.58, still above every matched competitor but farther from its lower 0.89 ceiling (Table 11).

MuJoCo: comparison by return. Figure 3 and Table 14 report the six MuJoCo families under the return-comparison rule against the framework's channel, the three same-trial heads, the hybrid, and IDAQ at five and ten episodes. VeriCode is at or above the best competitor in all eighteen cells, significantly above in thirteen and below in none, with a mean return of 267 against 200 for the strongest same-trial head and 170 for the hybrid. The two regimes gain for different reasons: on cheetah-vel and ant-goal the three same-trial heads are 14 to 140 return points below the encoder's oracle-context code and the rule empirically recovers much of that context term, while on hopper and walker the best training code is 23 to 140 above it and the rule recovers most of that latent ceiling. The committed candidate is spread over the whole set (likelihood 61 percent, exploration 23, prior 8, encoder 8), so no fixed head matches the aggregate decisions of the comparison rule.

7 Limitations and conclusion

The procedure is not free. It needs a success signal, or a return informative in one episode: removing the signal on Meta-World and committing by return reaches 0.71 against 0.52 for the hybrid, at 8.4 rather than 5.5 episodes (Table 11). It spends up to two (Meta-World) or five (MuJoCo) more episodes than the heads it compares. Proposition 3 applies directly only to fixed candidates and fresh independent episodes; repeatable success makes outcome reuse exact, but neither result covers the agreement shortcut or the full adaptive rule. With a stochastic actor the method keeps its lead but loses a quarter of its success. It gains least when an earlier component is already strong (plate-slide), when the candidate set is poor (cheetah-vel without the prior candidate), and in the 15 percent of Meta-World trials in which excitation and both heads fail together.

These results also delimit when the wrapper is appropriate. Before deployment, code-set rollouts on representative validation shifts can test whether the frozen policy contains useful candidates, while the decomposition can distinguish missing policy capability from a context or code-selection failure. A weak code-set ceiling points to retraining or a richer candidate generator, not a longer verification budget. The budget should also reflect the reliability of the available outcome signal: one episode is defensible only when success is repeatable, whereas stochastic resets or noisy returns require fresh repetitions and therefore higher cost. Finally, candidate evaluation scales with the number of codes retained; substantially larger task libraries will require screening or hierarchical search while preserving direct outcome checks for the final alternatives.

The paper establishes a way to ask where an OOD failure comes from before designing a remedy and, where only the code is wrong, a remedy that gives direct outcomes priority over predicted confidence. On six Meta-World families, VeriCode raises test-time success from 0.52 to 0.83 with a mean of 5.5 adaptation episodes; on MuJoCo it raises return from 200 to 267 with 9.4 episodes. It requires no backbone retraining or calibrated confidence threshold. Its cheapest component is absent from the baselines considered here: in our deterministic harness, a training code that succeeds during exploration is more reliable than a code re-inferred from that episode. This suggests a general design rule for test-time adaptation: treat inference as candidate generation, preserve measurements obtained during exploration, and spend new interactions only where they resolve a consequential choice.

AI USE STATEMENT

In this work, generative AI tools assisted with developing and correcting formal mathematical statements, assumptions, and proof arguments, and with line-by-line proof checks; implementing and debugging method, evaluation, analysis, and release code; literature search and bibliographic verification; checking reported results and their interpretation against stored artifacts; and drafting and editing the manuscript and release artifacts. They did not originate the research question, conceptual decomposition, method, hypotheses, or experimental design; were not used to execute or select research experiments or to generate synthetic or experimental datasets; and were not used for translation, dataset cleaning or reformatting, or qualitative or thematic analysis, which were not applicable. The authors reviewed all AI-assisted work, manually audited the code and outputs, checked references against original or authoritative sources, and revised the text. We take responsibility for the final content, including all text, claims, code, and other AI-assisted artifacts.

ETHICS STATEMENT

This work evaluates frozen checkpoints of publicly available offline meta-reinforcement learning algorithms in simulated locomotion and manipulation environments. It involves no human subjects or personally identifiable data. No human-derived or sensitive data are released; the supplement contains only method, evaluation, and data-collection code. We are not aware of harmful applications that the method enables beyond those already possible with the algorithms it wraps, and we declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

The evaluation protocol, OOD splits, and exclusion rule are specified in Section 4 and Appendix K. The assumptions of every theorem are stated with the theorem, and complete proofs are in Appendices A to C. Appendix K gives the checkpoint provenance, construction of the training codes, decoder architecture and training schedule, exploration and selection settings, baseline collection rules, IDAQ re-implementation, and seeds per cell. The appendix tables report the per-cell numbers behind the aggregate results, including the seed standard errors of Table 18. The supplementary code contains the data-collection pipeline and a minimal implementation of VeriCode; pretrained checkpoints and result artifacts are intentionally omitted. The main Meta-World campaign was run a second time end to end on the same checkpoints with fresh adaptation trials and differed by 0.002 ± 0.005, paired over the 108 checkpoints.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2017.

Anurag Ajay, Abhishek Gupta, Dibya Ghosh, Sergey Levine, and Pulkit Agrawal. Distributionally adaptive meta reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. doi: 10.52202/068431-1875.

Kavosh Asadi, Dipendra Misra, and Michael L. Littman. Lipschitz continuity in model-based reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 264–273. PMLR, 2018.

Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *Proceedings of the 23rd Conference on Learning Theory (COLT)*, pp. 41–53, 2010.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 4055–4065, 2017.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. ISBN 9780199535255. doi: 10.1093/acprof:oso/9780199535255.001.0001.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19):1832–1852, 2011. doi: 10.1016/j.tcs.2010.12.059.

Jung-Hoon Cho, Heling Zhang, Siqi Du, Roy Dong, and Cathy Wu. Formalizing task-space complexity for zero-shot generalization. In *Proceedings of the 8th Annual Learning for Dynamics and Control Conference (L4DC)*, volume 331 of *Proceedings of Machine Learning Research*, pp. 1212–1228. PMLR, 2026.

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 (NeurIPS)*, volume 34, pp. 4607–4618, 2021.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite Markov decision processes. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 162–169. AUAI Press, 2004.

Haotian Fu, Jiayu Yao, Omer Gottesman, Finale Doshi-Velez, and George Konidaris. Performance bounds for model and policy transfer in hidden-parameter MDPs. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=20gBzEzgtiI.

Yunkai Gao, Rui Zhang, Jiaming Guo, Fan Wu, Qi Yi, Shaohui Peng, Siming Lan, Ruizhi Chen, Zidong Du, Xing Hu, Qi Guo, Ling Li, and Yunji Chen. Context shift reduction for offline meta-reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. doi: 10.52202/075280-3505.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018.

Hongcai He, Zetao Zheng, Anjie Zhu, Deqiang Ouyang, and Jie Shao. Improving generalization in offline meta-reinforcement learning via cross-task contexts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21637–21644, 2026. doi: 10.1609/aaai.v40i26.39313.

Chengxing Jia, Fuxiang Zhang, Yi-Chen Li, Chen-Xiao Gao, Xu-Hui Liu, Lei Yuan, Zongzhang Zhang, and Yang Yu. Disentangling policy from offline task representation learning via adversarial data augmentation. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 944–953, 2024.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline RL? In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5084–5096. PMLR, 2021.

Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, pp. 267–274, 2002.

Jeongmo Kim, Yisak Park, Minung Kim, and Seungyul Han. Task-aware virtual training: Enhancing generalization in meta-reinforcement learning for out-of-distribution tasks. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30720–30748. PMLR, 2025.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76:201–264, 2023. doi: 10.1613/jair.1.14174.

Ksenia Konyushkova, Yutian Chen, Tom Le Paine, Çağlar Gülçehre, Cosmin Paduraru, Daniel J. Mankowitz, Misha Denil, and Nando de Freitas. Active offline policy selection. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 24631–24644, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 1179–1191. Curran Associates, Inc., 2020.

Jeongyeol Kwon, Yonathan Efroni, Constantine Caramanis, and Shie Mannor. RL for latent MDPs: Regret guarantees and a lower bound. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 24523–24534, 2021.

Erwan Lecarpentier, David Abel, Kavosh Asadi, Yuu Jinnai, Emmanuel Rachelson, and Michael L. Littman. Lipschitz lifelong reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8270–8278, 2021. doi: 10.1609/aaai.v35i9.17006.

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 (ICLR)*, 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 (NeurIPS)*, volume 37, 2024. doi: 10.52202/079017-2408.

Sen Lin, Jialin Wan, Tengyu Xu, Yingbin Liang, and Junshan Zhang. Model-based offline meta-reinforcement learning with regularization. In *International Conference on Learning Representations (ICLR)*, 2022.

Evan Zheran Liu, Aditi Raghunathan, Percy Liang, and Chelsea Finn. Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6925–6935. PMLR, 2021.

Lianghui Liu and Zongzhang Zhang. Meta-normalizing flow for data-limited offline meta-reinforcement learning (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 41278–41280, 2026. doi: 10.1609/aaai.v40i48.42241.

Xingtu Liu. An information-theoretic analysis of OOD generalization in meta-reinforcement learning. *arXiv preprint arXiv:2510.23448*, 2025. doi: 10.48550/arXiv.2510.23448. URL https://arxiv.org/abs/2510.23448.

Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall PTR, Upper Saddle River, NJ, 2nd edition, 1999. ISBN 0136566952.

Long Ma, Fangwei Zhong, and Yizhou Wang. Behavior-agnostic task inference for robust offline in-context reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 42293–42308. PMLR, 2025.

Russell Mendonca, Xinyang Geng, Chelsea Finn, and Sergey Levine. Meta-reinforcement learning robust to distributional shift via model identification and experience relabeling. *arXiv preprint arXiv:2006.07178*, 2020.

Eric Mitchell, Rafael Rafailov, Xue Bin Peng, Sergey Levine, and Chelsea Finn. Offline meta-reinforcement learning with advantage weighting. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7780–7791. PMLR, 2021.

Mohammadreza Nakhaei, Aidan Scannell, Kevin Luck, and Joni Pajarinen. Temporal consistency improves generalization in contextual offline meta reinforcement learning. *arXiv preprint arXiv:2603.02935*, 2026. doi: 10.48550/arXiv.2603.02935. URL https://arxiv.org/abs/2603.02935.

Mohammadreza Nakhaeinezhadfard, Aidan Scannell, and Joni Pajarinen. Entropy regularized task representation learning for offline meta-reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19616–19623, 2025. doi: 10.1609/aaai.v39i18.34160.

Kumpati S. Narendra and Anuradha M. Annaswamy. Persistent excitation in adaptive systems. *International Journal of Control*, 45(1):127–160, 1987. doi: 10.1080/00207178708933715.

Vitchyr H. Pong, Ashvin V. Nair, Laura M. Smith, Catherine Huang, and Sergey Levine. Offline meta-reinforcement learning with online self-supervision. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17811–17829. PMLR, 2022.

Fuyuan Qian, Menglong Zhang, Song Wang, and Quanying Liu. Behavior-invariant task representation learning with transformer-based world models for offline meta-reinforcement learning. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://arxiv.org/abs/2606.00780.

Emmanuel Rachelson and Michail G. Lagoudakis. On the locality of action domination in sequential decision making. In *Proceedings of the 11th International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, Fort Lauderdale, Florida, USA, 2010.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5331–5340. PMLR, 2019.

Thomas Rupf, Marco Bagatella, Marin Vlastelica, and Andreas Krause. Optimistic task inference for behavior foundation models. In *International Conference on Learning Representations*, 2026.

Luca Sabbioni, Francesco Corda, and Marcello Restelli. Stepsize learning for policy gradient methods in contextual Markov decision processes. In *Machine Learning and Knowledge Discovery in Databases: Research Track (ECML PKDD)*, volume 14172 of *Lecture Notes in Computer Science*, pp. 506–523. Springer, 2023. doi: 10.1007/978-3-031-43421-1_30.

Shagun Sodhani, Franziska Meier, Joelle Pineau, and Amy Zhang. Block contextual MDPs for continual learning. In *Proceedings of the 4th Annual Learning for Dynamics and Control Conference (L4DC)*, volume 168 of *Proceedings of Machine Learning Research*, pp. 608–623. PMLR, 2022.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Jianhao Wang, Jin Zhang, Haozhe Jiang, Junyu Zhang, Liwei Wang, and Chongjie Zhang. Offline meta reinforcement learning with in-distribution online adaptation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36626–36669. PMLR, 2023.

Min Wang, Xin Li, Mingzhong Wang, and Hasnaa Bennis. Offline meta-reinforcement learning with flow-based task inference and adaptive correction of feature overgeneralization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26390–26397, 2026. doi: 10.1609/aaai.v40i31.39845.

Zhi Wang, Li Zhang, Wenhao Wu, Yuanheng Zhu, Dongbin Zhao, and Chunlin Chen. Meta-DT: Offline meta-RL as conditional sequence modeling with world model disentanglement. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. doi: 10.52202/079017-1425.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

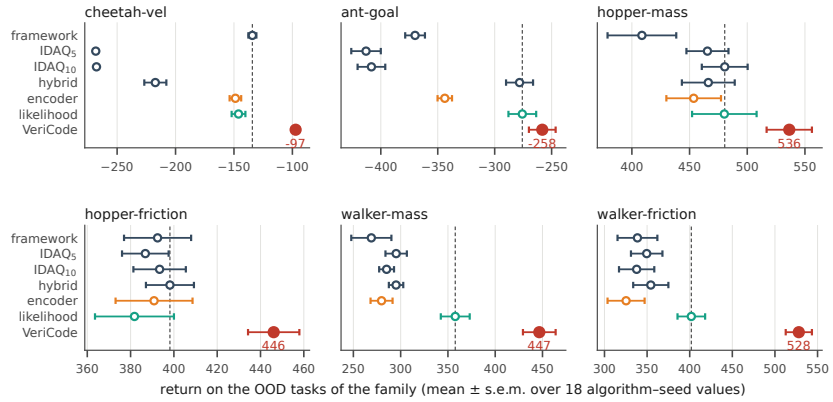


Figure 3: **MuJoCo, return by environment.** Return on each family's OOD tasks, mean ± s.e.m. over 18 algorithm–seed values (three algorithms × six seeds); the dashed line is the best family-level competitor. The encoder and likelihood heads are recorded in the same trials as VeriCode; the framework, hybrid, and IDAQ baselines are separate campaigns on the same frozen checkpoints. VeriCode commits to the candidate with the largest one-episode comparison return. Table 14 gives per-cell values.

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 *Proceedings of the Conference on Robot Learning (CoRL)*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, 2020.

Haoqi Yuan and Zongqing Lu. Robust task representations for offline meta-reinforcement learning via contrastive learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 25747–25759. PMLR, 2022.

Amy Zhang, Shagun Sodhani, Khimya Khetarpal, and Joelle Pineau. Learning robust state abstractions for hidden-parameter block MDPs. In *International Conference on Learning Representations (ICLR)*, 2021.

Hai Zhang, Boyuan Zheng, Tianying Ji, Jinhang Liu, Anqi Guo, Junqiao Zhao, and Lanqing Li. Scrutinize what we ignore: Reining in task representation shift of context-based offline meta reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025a.

Menglong Zhang, Fuyuan Qian, and Quanying Liu. Learning task belief similarity with latent dynamics for meta-reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025b.

Renzhe Zhou, Chen-Xiao Gao, Zongzhang Zhang, and Yang Yu. Generalizable task representation learning for offline meta-reinforcement learning with data limitations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17132–17140, 2024. doi: 10.1609/aaai.v38i15.29658.

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 Representations (ICLR)*, 2020.

A Proofs for Section 3

A.1 Conventions

Total variation is $\mathrm{TV}(\mu,\nu) = \sup_A |\mu(A) - \nu(A)| = \frac{1}{2}\|\mu - \nu\|_1$. For a bounded measurable f and any constant c, $\int f\, d(\mu - \nu) = \int (f - c)\, d(\mu - \nu) \le \|f - c\|_\infty \|\mu - \nu\|_1$; taking c to be the midpoint of the range of f gives the centering inequality $|\int f\, d\mu - \int f\, d\nu| \le \mathrm{TV}(\mu,\nu)\,\mathrm{span}(f)$.

Values and action values lie in $[r_{\min}/(1-\gamma), r_{\max}/(1-\gamma)]$, so their span is at most $\Delta_r/(1-\gamma)$. A policy is any measurable map from histories $h_t = (s_0, a_0, \ldots, s_t)$ to action distributions; π_z is the stationary Markov policy $\pi(\cdot \mid s, z)$ with z frozen, and $\mu_z(s) = \tanh(m(s, z))$ its deterministic version. Code norms are Euclidean. Meta-World rewards lie in $[0, 10]$, so $\Delta_r = 10$ there. We use one fact about sequential laws repeatedly: if two laws of a trajectory segment share the initial distribution and the policy kernels and differ only in k applications of the transition kernel, then their total variation is at most $k \sup_{(s,a)\in\mathcal{K}} \mathrm{TV}(P_\theta, P_{\theta'})$. This follows by writing the difference as a telescoping sum of k hybrid laws, consecutive hybrids differing in a single kernel application, and bounding each difference by the centering inequality applied conditionally on the past; since $\mathcal{K}$ contains the supports of both occupancies, the state-action pair at which the kernels are compared lies in it.

A.2 Proof of Lemma 1

Fix a policy π and two tasks θ, θ'. Let $\mathbb{P}_\theta^{(t)}$ denote the law of $(s_0, a_0, \ldots, s_t, a_t, s_{t+1})$ under (ρ, π, P_θ). The two laws differ in $t+1$ kernel applications, so their total variation is at most $(t+1)\sup_{(s,a)\in\mathcal{K}} \mathrm{TV}(P_\theta, P_{\theta'})$ by the telescoping fact of Appendix A.1. Since rewards are bounded we may exchange sum and expectation. This lets us write

$$J_\theta(\pi) - J_{\theta'}(\pi) = \sum_{t\geq 0} \gamma^t \Big\{ \mathbb{E}_\theta\big[(r_\theta - r_{\theta'})(s_t, a_t, s_{t+1})\big] + \big(\mathbb{E}_\theta - \mathbb{E}_{\theta'}\big)\big[r_{\theta'}(s_t, a_t, s_{t+1})\big] \Big\}.$$

The first bracket is at most $\sup_{(s,a)\in\mathcal{K}} \sup_{s'} |r_\theta - r_{\theta'}|$ because $(s_t, a_t) \in \mathcal{K}$ almost surely. The second bracket is the difference of the expectations of the bounded function $r_{\theta'}$ under the two laws, so by centering it is at most $(t+1)\Delta_r \sup_{(s,a)\in\mathcal{K}} \mathrm{TV}$. Summing with $\sum_t \gamma^t = (1-\gamma)^{-1}$ and $\sum_t \gamma^t (t+1) = (1-\gamma)^{-2}$ gives $J_\theta(\pi) - J_{\theta'}(\pi) \leq (1-\gamma)^{-1} \sup|\Delta r| + (1-\gamma)^{-2}\Delta_r \sup \mathrm{TV} = d(\theta, \theta')/(1-\gamma)$, and the other direction is symmetric. If rewards depend only on (s, a), the relevant law is that of $(s_0, a_0, \ldots, s_t, a_t)$, which involves t kernel applications, and $\sum_t \gamma^t t = \gamma(1-\gamma)^{-2}$ gives the improved coefficient $\gamma\Delta_r/(1-\gamma)$. For the optimal returns, pick an ϵ-optimal policy π for θ; then $J^*(\theta) - J^*(\theta') \leq J_\theta(\pi) + \epsilon - J_{\theta'}(\pi) \leq d(\theta, \theta')/(1-\gamma) + \epsilon$, let $\epsilon \downarrow 0$ and symmetrise. □

Remark 1 (Scope and tightness of Lemma 1). Nothing beyond Lemma 1 is needed for optimal returns as long as it is stated for the full policy class, which is why we allowed history-dependent policies. A code-conditioned policy is covered if and only if the code is the same frozen vector in both tasks; the evaluation harness infers the code once and then acts, so this holds for every return we report. The supremum in d may be restricted to the union of the supports of the two occupancies. The bound is first-order tight. Consider a two-state chain with rewards in $[0, r_{\max}]$, initially in a live state, in which a transition remains live with probability $1-\eta$ and earns $r_{\max}$ exactly when its next state is live; otherwise it enters an absorbing terminal state and earns zero. The return with $\eta = 0$ minus the return with absorption probability η equals $r_{\max}\eta/((1-\gamma)(1-\gamma+\gamma\eta))$. The ratio of this difference to the bound $\Delta_r\eta/(1-\gamma)^2$ tends to one as $\eta \downarrow 0$.

A.3 Lemma 1-W: the Wasserstein form for deterministic simulators

Let $\rho_{\mathcal{S}}$ be a metric on $\mathcal{S}$ and W_1 the Wasserstein-1 distance it induces. MuJoCo kernels are Dirac, $P_\theta(\cdot \mid s, a) = \delta_{f_\theta(s,a)}$, so $\mathrm{TV}(P_\theta, P_{\theta'})(s, a) = 1$ whenever $f_\theta(s, a) \neq f_{\theta'}(s, a)$ and equation 1 is vacuous for dynamics-shift pairs. The following lemma replaces total variation by transport cost in the manner of Asadi et al. (2018). Fix a common transition coefficient $c_P \geq 0$ and define

$$d_{\mathrm{W}}(\theta, \theta') := \sup_{(s,a)\in\mathcal{K}} \sup_{s'} |r_\theta(s, a, s') - r_{\theta'}(s, a, s')| + c_P \sup_{(s,a)\in\mathcal{K}} W_1\big(P_\theta(\cdot \mid s, a), P_{\theta'}(\cdot \mid s, a)\big). \quad (5)$$

Lemma 1-W (Values are Lipschitz in the Wasserstein task metric)**.** *Let π be a stationary Markov policy. Suppose (W1) $s' \mapsto r_{\theta'}(s, a, s')$ is L_r-Lipschitz for every $(s, a) \in \mathcal{K}$, (W2) $V_{\theta'}^\pi$ is L_V-Lipschitz on $\mathcal{S}$, and $c_P \geq L_r + \gamma L_V$. Then $|J_\theta(\pi) - J_{\theta'}(\pi)| \leq d_{\mathrm{W}}(\theta, \theta')/(1-\gamma)$. For Dirac kernels, $W_1(P_\theta, P_{\theta'})(s, a) = \rho_{\mathcal{S}}(f_\theta(s, a), f_{\theta'}(s, a))$. The same bound holds for $|J^*(\theta) - J^*(\theta')|$ if, for every $\epsilon > 0$, both comparison directions admit stationary ϵ-optimal policies for which (W1)–(W2) hold in the destination task with the same coefficient c_P.*

Proof. Write $(T_\theta g)(s, a) := \int P_\theta(ds' \mid s, a)[r_\theta(s, a, s') + \gamma g(s')]$ and $T_\theta^\pi g(s) := \int \pi(da \mid s)(T_\theta g)(s, a)$, so that $V_\theta^\pi = T_\theta^\pi V_\theta^\pi$. Then $V_\theta^\pi - V_{\theta'}^\pi = \gamma P_\theta^\pi (V_\theta^\pi - V_{\theta'}^\pi) + (T_\theta^\pi - T_{\theta'}^\pi)V_{\theta'}^\pi$, and

solving the linear equation and integrating against ρ gives

$$J_\theta(\pi) - J_{\theta'}(\pi) = \frac{1}{1-\gamma}\,\mathbb{E}_{(s,a)\sim d_\theta^\pi}\Big[(T_\theta - T_{\theta'})V_{\theta'}^\pi(s,a)\Big],$$

where d_θ^π is the normalised discounted state-action occupancy. The integrand splits as $\int P_\theta(ds')(r_\theta - r_{\theta'})(s,a,s') + \int (P_\theta - P_{\theta'})(ds')[r_{\theta'}(s,a,s') + \gamma V_{\theta'}^\pi(s')]$. The absolute value of the first part is bounded by the reward term of equation 5. In the second part, the integrand $s' \mapsto r_{\theta'}(s,a,s') + \gamma V_{\theta'}^\pi(s')$ is $(L_r + \gamma L_V)$-Lipschitz, so Kantorovich–Rubinstein duality bounds its absolute value by $c_P W_1(P_\theta(\cdot \mid s,a), P_{\theta'}(\cdot \mid s,a))$. Taking the absolute value of the occupancy expectation proves the fixed-policy result. The optimal-return claim follows by applying this result to the stipulated ϵ-optimal policies in each direction and taking $\epsilon \downarrow 0$. □

Remark 2 (Reading Lemma 1-W). For Dirac kernels, define the realised reward $\widetilde{r}_\theta(s,a) := r_\theta(s,a,f_\theta(s,a))$. Repeating the proof with $V_{\theta'}^\pi$ as the destination-task reference gives the sharper pointwise pseudometric

$$d_{\mathrm{W}}^{\mathrm{pt}}(\theta,\theta') := \sup_{(s,a)\in\mathcal{K}} \big\{ |\widetilde{r}_\theta(s,a) - \widetilde{r}_{\theta'}(s,a)| + c_P \rho_{\mathcal{S}}\big(f_\theta(s,a), f_{\theta'}(s,a)\big) \big\}. \tag{6}$$

Indeed, the Bellman integrand is $\widetilde{r}_\theta - \widetilde{r}_{\theta'} + \gamma(V_{\theta'}^\pi \circ f_\theta - V_{\theta'}^\pi \circ f_{\theta'})$, whose absolute value is bounded by the expression inside braces in equation 6 whenever $V_{\theta'}^\pi$ is L_V-Lipschitz and $c_P \geq \gamma L_V$. Hence $|J_\theta(\pi) - J_{\theta'}(\pi)| \leq d_{\mathrm{W}}^{\mathrm{pt}}(\theta,\theta')/(1-\gamma)$. The optimal-return version requires the same common c_P for stationary ϵ-optimal policies in both comparison directions, as in Lemma 1-W. This pointwise form, which is no larger than the corresponding sum of separate suprema, is the metric used in Theorem 2-A. The realised-reward difference and next-state displacement are what we measure for the mass and friction families (Appendix H.1), but the empirical displacement is only a proxy until a valid common c_P is supplied. If the same route uses total variation, the span of $r_{\theta'} + \gamma V_{\theta'}^\pi$ is at most $\Delta_r/(1-\gamma)$, recovering Lemma 1. The L_V condition is substantive: it follows under suitable Lipschitz rewards, dynamics, and policies, for example when $\gamma L_f(1 + L_{\pi,s}) < 1$ (Rachelson & Lagoudakis, 2010), but MuJoCo does not guarantee it.

A.4 Lemma 2 and its proof

Lemma 2 (Performance difference under policy discrepancy)**.** *Fix a task θ and let d_θ^π be the normalised discounted state occupancy of π. (a) For stochastic Markov policies π, π', $|J_\theta(\pi) - J_\theta(\pi')| \leq \frac{\Delta_r}{(1-\gamma)^2}\,\mathbb{E}_{s\sim d_\theta^\pi}\mathrm{TV}(\pi(\cdot \mid s), \pi'(\cdot \mid s)) \leq \frac{\Delta_r}{(1-\gamma)^2}\sup_s \mathrm{TV}(\pi(\cdot \mid s), \pi'(\cdot \mid s))$. (b) For deterministic policies $a = \mu(s)$ and $a = \mu'(s)$, if $a \mapsto Q_\theta^{\mu'}(s,a)$ is L_Q-Lipschitz for $s \in \operatorname{supp} d_\theta^\mu$, then $|J_\theta(\mu) - J_\theta(\mu')| \leq \frac{L_Q}{1-\gamma}\,\mathbb{E}_{s\sim d_\theta^\mu}\|\mu(s) - \mu'(s)\|_2 \leq \frac{L_Q}{1-\gamma}\sup_s \|\mu(s) - \mu'(s)\|_2$.*

Part (a) is the performance difference lemma of Kakade & Langford (2002) followed by centering, which is why the constant is $\Delta_r/(1-\gamma)^2$ rather than the $2\Delta_r/(1-\gamma)^2$ that a Hölder step would give (Achiam et al., 2017). The action displacement in (b) is exactly what the sensitivity probe measures.

Part (a). Fix θ and drop it from the notation. For any two policies, the telescoping identity $\mathbb{E}^\pi \sum_t \gamma^t[\gamma V^{\pi'}(s_{t+1}) - V^{\pi'}(s_t)] = -J(\pi')$ and the tower property $\mathbb{E}[r(s_t,a_t,s_{t+1}) + \gamma V^{\pi'}(s_{t+1}) \mid s_t, a_t] = Q^{\pi'}(s_t,a_t)$ give the performance difference lemma

$$\begin{aligned} J(\pi) - J(\pi') &= \mathbb{E}^\pi \sum_t \gamma^t \big[Q^{\pi'}(s_t,a_t) - V^{\pi'}(s_t)\big] \\ &= \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d^\pi} \int Q^{\pi'}(s,a)\big(\pi(da \mid s) - \pi'(da \mid s)\big), \end{aligned} \tag{7}$$

where we used $V^{\pi'}(s) = \int Q^{\pi'}(s,a)\pi'(da \mid s)$; the identity holds for s'-dependent rewards because Q absorbs the next state. For each s the inner integral is the difference of the expectations of $Q^{\pi'}(s,\cdot)$ under two action distributions, so the centering inequality with $\operatorname{span}_a Q^{\pi'}(s,\cdot) \leq \Delta_r/(1-\gamma)$ establishes part (a). A direct Hölder step using $|A^{\pi'}| \leq \Delta_r/(1-\gamma)$ and $\|\pi - \pi'\|_1 = 2\mathrm{TV}(\pi,\pi')$ would give $2\Delta_r/(1-\gamma)^2$ instead.

Part (b). For deterministic policies $V^{\mu'}(s) = Q^{\mu'}(s, \mu'(s))$, so equation 7 becomes the exact identity $J(\mu) - J(\mu') = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\mu}}[Q^{\mu'}(s, \mu(s)) - Q^{\mu'}(s, \mu'(s))]$, and the Lipschitz assumption on $a \mapsto Q^{\mu'}(s, a)$ for $s \in \operatorname{supp} d^{\mu}$ yields part (b). The factor $1/(1-\gamma)$ appears once; the other horizon factor sits inside L_Q.

Part (c), the bridge. If evaluation samples $a = \tanh(u)$ with $u \sim \mathcal{N}(m(s, z), \sigma^2 I)$, then, because a coordinatewise bijection leaves total variation unchanged and two Gaussians with a common covariance differ along one direction, $\mathrm{TV}(\pi(\cdot \mid s, z), \pi(\cdot \mid s, z')) = 2\Phi(\|m(s, z) - m(s, z')\|/(2\sigma)) - 1 \le \|m(s, z) - m(s, z')\|/(\sigma\sqrt{2\pi})$. More generally, for a common diagonal covariance whose standard deviations are at least $\sigma_{\min}$, whitening gives $L_\pi \le L_m/(\sigma_{\min}\sqrt{2\pi})$ when m is L_m-Lipschitz in the code, while $L_\mu \le L_m$ since $\tanh$ is 1-Lipschitz. The stochastic constant diverges as $\sigma_{\min} \to 0$, so (b) is the only route for deterministic evaluation. □

A.5 Assumptions and proof of Theorem 1

The three quantities of Section 3 enter as the following assumptions:

Assumption 1 (Offline gap on a training task)**.** $\varepsilon_i := J^*(\theta_i) - J_{\theta_i}(\pi^{\mathrm{ev}}_{\bar{z}_i}) \ge 0$ for the fixed code $\bar{z}_i$ and the chosen evaluation mode.

Assumption 2 (Policy sensitivity to the code, for the pair compared)**.** Either $\sup_s \mathrm{TV}(\pi(\cdot \mid s, \bar{z}_i), \pi(\cdot \mid s, \hat{z})) \le L_\pi \|\hat{z} - \bar{z}_i\|_2$ (stochastic evaluation) or $\sup_s \|\mu(s, \bar{z}_i) - \mu(s, \hat{z})\|_2 \le L_\mu \|\hat{z} - \bar{z}_i\|_2$ (deterministic evaluation).

Assumption 3 (Action-Lipschitz critic for deterministic evaluation)**.** $a \mapsto Q^{\mu_{\bar{z}_i}}_{\theta^*}(s, a)$ is L_Q-Lipschitz for $s \in \operatorname{supp} d^{\mu_{\hat{z}}}_{\theta^*}$.

Assumption 2 is only needed for the single pair $(\bar{z}_i, \hat{z})$ that the theorem compares, not on a region. The sensitivity probe provides a sampled diagnostic for that pair; the learned critic likewise gives a diagnostic proxy for L_Q, not the constant itself.

Adding and subtracting $J^*(\theta_i)$, $J_{\theta_i}(\pi^{\mathrm{ev}}_{\bar{z}_i})$, and $J_{\theta^*}(\pi^{\mathrm{ev}}_{\bar{z}_i})$ yields equation 2 for every i and $\hat{z}$ with no assumption. Lemma 1 applied to the optimal returns bounds $|A_i|$, and applied to the fixed evaluation policy $\pi^{\mathrm{ev}}_{\bar{z}_i}$ in θ_i and θ^* bounds $|B_i|$, both by $d(\theta^*, \theta_i)/(1-\gamma)$. For dynamics shifts, these steps use Lemma 1-W, or its pointwise realised-reward form equation 6, only under the common-coefficient conditions stated in Theorem 1: the fixed evaluation policy suffices for B_i, while A_i additionally requires the condition for the near-optimal policies in both comparison directions. For stochastic evaluation, apply Lemma 2(a) with $\pi = \pi_{\hat{z}}$ and $\pi' = \pi_{\bar{z}_i}$. For deterministic evaluation, apply Lemma 2(b) with $\mu = \mu_{\hat{z}}$ and $\mu' = \mu_{\bar{z}_i}$; the occupancy is $d^{\mu_{\hat{z}}}_{\theta^*}$ and the critic is $Q^{\mu_{\bar{z}_i}}_{\theta^*}$, exactly as in Assumption 3. Assumption 2 then gives the two bounds in equation 3. Minimising the applicable bound over i completes the proof. □

Remark 3 (Endpoint versus segment). Only the discrepancy between the two endpoint evaluation policies $\pi^{\mathrm{ev}}_{\bar{z}_i}$ and $\pi^{\mathrm{ev}}_{\hat{z}}$ enters, on states from actual rollouts of $\pi^{\mathrm{ev}}_{\hat{z}}$ in θ^*; this is the sensitivity probe's endpoint measurement. In the deterministic branch, if differentiability of $z \mapsto \mu_z(s)$ is available only along the segment $z(t) = \bar{z}_i + t(\hat{z} - \bar{z}_i)$, the endpoint action displacement is at most the path length, which is at most $L_{\mathrm{seg}}\|\hat{z} - \bar{z}_i\|$ with L_{seg} the largest directional derivative along the segment.

A.6 The reference-policy version

Let π^E_θ be the data-collecting expert of task θ and $E(\theta) := J_\theta(\pi^E_\theta) \le J^*(\theta)$. The MuJoCo decomposition uses E in place of J^*; the Meta-World decomposition uses the common success ceiling 1 descriptively rather than a data-collecting expert return.

Proposition A.1 (Decomposition with experts in place of J^*)**.** *For every i and $\hat{z}$,*

$$E(\theta^*) - J_{\theta^*}(\pi^{\mathrm{ev}}_{\hat{z}}) = A^E_i + \varepsilon^E_i + B_i + C_i(\hat{z}), \qquad A^E_i := E(\theta^*) - E(\theta_i), \quad \varepsilon^E_i := E(\theta_i) - J_{\theta_i}(\pi^{\mathrm{ev}}_{\bar{z}_i}),$$

with B_i, C_i as in equation 2, and $A^E_i \le d(\theta^*, \theta_i)/(1-\gamma) + \kappa_i$ where $\kappa_i := J_{\theta_i}(\pi^E_{\theta^*}) - J_{\theta_i}(\pi^E_{\theta_i})$ is the expert cross-task regret. Consequently $E(\theta^*) - J_{\theta^*}(\pi^{\mathrm{ev}}_{\hat{z}}) \le 2d(\theta^*, \theta_i)/(1-\gamma) + \kappa_i + \varepsilon^E_i + \bar{C}_i(\hat{z})$. Alternatively, if $J^*(\theta_i) - E(\theta_i) \le \delta_E$, then $E(\theta^*) - J_{\theta^*}(\pi^{\mathrm{ev}}_{\hat{z}}) \le 2d(\theta^*, \theta_i)/(1-\gamma) + \varepsilon^E_i + \delta_E + \bar{C}_i(\hat{z})$. The corresponding dynamics-shift statements require the common-coefficient conditions of Lemma 1-W.

Proof. The identity is the same telescoping. For the bound on A_i^E write $E(\theta^*) - E(\theta_i) = [J_{\theta^*}(\pi^E_{\theta^*}) - J_{\theta_i}(\pi^E_{\theta^*})] + [J_{\theta_i}(\pi^E_{\theta^*}) - J_{\theta_i}(\pi^E_{\theta_i})]$; the first bracket is Lemma 1 for the fixed policy $\pi^E_{\theta^*}$ and the second is κ_i. For the alternative, the measured gap is at most the true gap because $E(\theta^*) \le J^*(\theta^*)$, the true gap obeys Theorem 1, and the true ε_i is at most $\varepsilon_i^E + \delta_E$. □

The regret κ_i is measurable with one extra rollout family, the θ^*-expert acting on θ_i; it is nonpositive exactly when the θ_i-expert beats the θ^*-expert on its own task, in which case the κ_i correction may be dropped. This does not by itself certify a metric proxy or the global constants in $\bar{C}_i$. Without either κ_i or a uniform δ_E, nothing guarantees that the measured gap is at most the corresponding right-hand side. On cheetah-vel the measured κ_i is negative on every OOD task, so no positive expert-regret correction is needed; on the hopper families it is positive on the light or low-friction side, so the κ_i term is needed. The Meta-World success decomposition instead uses the common ceiling 1 as a descriptive reference and makes no expert-regret bound claim (Table 7).

A.7 Finite-horizon undiscounted version

The experiments report undiscounted returns over $H = 200$ steps. Let

$$J_\theta^H(\pi) := \mathbb{E}^{\pi,\theta} \sum_{t=0}^{H-1} r_\theta(s_t, a_t, s_{t+1}), \qquad J_H^*(\theta) := \sup_\pi J_\theta^H(\pi),$$

with the absorbing termination state as before. The time-t action value Q_t^π has span at most $(H - t)\Delta_r$. Define the finite-horizon metric

$$d_H(\theta, \theta') := \sup_{(s,a)\in\mathcal{K}} \sup_{s'} |r_\theta - r_{\theta'}| + \frac{H+1}{2} \Delta_r \sup_{(s,a)\in\mathcal{K}} \mathrm{TV}(P_\theta, P_{\theta'}), \tag{8}$$

so that $H\, d_H = H \sup|\Delta r| + \binom{H+1}{2}\Delta_r \sup \mathrm{TV}$. For reward-shift families $d_H = \sup|\Delta r|$; for the velocity family, $H\, d_H \le 200|v - v'|$, with equality when $\mathcal{K}$ contains a speed outside the interval between the two targets.

Lemma 1-H. *For every policy π, $|J_\theta^H(\pi) - J_{\theta'}^H(\pi)| \le H \sup|\Delta r| + \binom{H+1}{2}\Delta_r \sup \mathrm{TV} = H\, d_H(\theta, \theta')$, and the same bound holds for J_H^*. If rewards depend only on (s, a), $\binom{H+1}{2}$ improves to $\binom{H}{2}$. The Wasserstein statements that follow take π to be a time-dependent Markov policy. Under (W1) and (W2) for the time-t values V_t^π with constants $L_{V,t}$, $|J_\theta^H(\pi) - J_{\theta'}^H(\pi)| \le H\big(\sup|\Delta r| + (L_r + \bar{L}_V) \sup W_1\big)$ with $\bar{L}_V := \frac{1}{H}\sum_{t<H} L_{V,t+1}$. For deterministic kernels and realised rewards, define*

$$d^{\mathrm{pt}}_{\mathrm{W},H}(\theta, \theta') := \sup_{(s,a)\in\mathcal{K}} \{|\widetilde{r}_\theta - \widetilde{r}_{\theta'}| + c_{P,H}\rho_{\mathcal{S}}(f_\theta, f_{\theta'})\}.$$

If $c_{P,H} \ge \max_{t<H} L_{V,t+1}$, then $|J_\theta^H(\pi) - J_{\theta'}^H(\pi)| \le H d^{\mathrm{pt}}_{\mathrm{W},H}(\theta, \theta')$. The corresponding optimal-return claim requires, for every $\epsilon > 0$ and in both comparison directions, time-dependent Markov ϵ-optimal policies whose destination-task values obey these Lipschitz bounds with the same $c_{P,H}$.

Proof. Identical to Lemma 1 with $\gamma = 1$ and the sums truncated: $\sum_{t=0}^{H-1} 1 = H$ and $\sum_{t=0}^{H-1}(t+1) = \binom{H+1}{2}$; for (s, a)-rewards $\sum_{t=0}^{H-1} t = \binom{H}{2}$. The Wasserstein form follows from the finite-horizon Bellman recursion $V_t = T_\theta V_{t+1}$ with $V_H \equiv 0$, telescoped as in Lemma 1-W, where the step-t integrand is $(L_r + L_{V,t+1})$-Lipschitz. For deterministic kernels, the realised-reward step-t integrand is bounded pointwise by $|\widetilde{r}_\theta - \widetilde{r}_{\theta'}| + L_{V,t+1}\rho_{\mathcal{S}}(f_\theta, f_{\theta'})$, giving the stated bound after summing H terms. The optimal-return extension uses the same ϵ-optimal-policy argument as Lemma 1-W. □

Lemma 2-H. *(a) For stochastic Markov policies,*

$$|J^H(\pi) - J^H(\pi')| \le \Delta_r \sum_{t<H} (H - t)\, \mathbb{E}^\pi \mathrm{TV}(\pi, \pi')(s_t) \le \binom{H+1}{2} \Delta_r \sup_s \mathrm{TV}(\pi(\cdot \mid s), \pi'(\cdot \mid s)).$$

(b) For deterministic policies, $J^H(\mu) - J^H(\mu') = \sum_{t<H} \mathbb{E}^\mu[Q_t^{\mu'}(s_t, \mu(s_t)) - Q_t^{\mu'}(s_t, \mu'(s_t))]$, and if $a \mapsto Q_t^{\mu'}(s, a)$ is $L_{Q,t}$-Lipschitz then, with $\bar{L}_Q := \frac{1}{H}\sum_{t<H} L_{Q,t}$, $|J^H(\mu) - J^H(\mu')| \le H\bar{L}_Q \sup_s \|\mu(s) - \mu'(s)\|_2$.

Proof. The finite-horizon performance difference lemma

$$J^H(\pi) - J^H(\pi') = \sum_{t<H} \mathbb{E}^{\pi} \int Q_t^{\pi'}(s_t, a)\big(\pi(da \mid s_t) - \pi'(da \mid s_t)\big)$$

follows from the telescoping of $V_t^{\pi'}$ with $V_H^{\pi'} \equiv 0$; centering with $\operatorname{span} Q_t^{\pi'} \le (H-t)\Delta_r$ and $\sum_{t=0}^{H-1}(H-t) = \binom{H+1}{2}$ give (a), and the Dirac case gives (b). □

Theorem 1-H (Finite-horizon decomposition)**.** *With the brackets of equation 2 defined through J^H, J_H^*, and the evaluation policy π_z^{ev}, the identity $J_H^*(\theta^*) - J_{\theta^*}^H(\pi_{\hat{z}}^{\mathrm{ev}}) = A_i + \varepsilon_i^H + B_i + C_i^H(\hat{z})$ holds. Under the finite-horizon analogues of Assumptions 2 and 3, $|A_i|, |B_i| \le H\, d_H(\theta^*, \theta_i)$ and $C_i^H(\hat{z}) \le \binom{H+1}{2}\Delta_r L_\pi \|\hat{z} - \bar{z}_i\|$ for stochastic evaluation, or $C_i^H(\hat{z}) \le H\bar{L}_Q L_\mu \|\hat{z} - \bar{z}_i\|$ for deterministic evaluation. Consequently $J_H^*(\theta^*) - J_{\theta^*}^H(\pi_{\hat{z}}^{\mathrm{ev}}) \le \min_i\{2H\, d_H(\theta^*, \theta_i) + \varepsilon_i^H + \bar{C}_i^H(\hat{z})\}$.*

The proof is that of Theorem 1 with Lemmas 1-H and 2-H. Table 2 lists the constants side by side; the finite-horizon transition constants are about double the discounted ones at $\gamma = 0.99$, and H^2 in place of $\binom{H+1}{2}$ would be loose by a further factor of about two.

Table 2: Horizon constants in the two settings: discounted with $\gamma = 0.99$ against undiscounted with $H = 200$, as they appear in Lemmas 1, 1-H, 2 and 2-H.

coefficient of	discounted, $\gamma = 0.99$	finite horizon, $H = 200$
$\sup\lvert\Delta r\rvert$ (Lemma 1)	$1/(1-\gamma) = 100$	$H = 200$
$\Delta_r \sup \mathrm{TV}$, rewards $r(s,a)$	$\gamma/(1-\gamma)^2 = 9{,}900$	$\binom{H}{2} = 19{,}900$
$\Delta_r \sup \mathrm{TV}$, rewards $r(s,a,s')$	$1/(1-\gamma)^2 = 10{,}000$	$\binom{H+1}{2} = 20{,}100$
$\Delta_r \sup_s \mathrm{TV}(\pi, \pi')$ (Lemma 2a)	$1/(1-\gamma)^2 = 10{,}000$	$\binom{H+1}{2} = 20{,}100$
$L_Q \sup_s \lVert\mu - \mu'\rVert$ (Lemma 2b)	$1/(1-\gamma) = 100$	$H = 200$ (with $\bar{L}_Q$)

Remark 4 (Numerical check of the constants). We checked the finite-horizon constants numerically: on 3,000 random tabular MDP pairs the bound of Lemma 1-H was never violated, and on the absorbing chain of Remark 1 with $H = 200$ and a per-step transition difference of 10^{-4} the ratio of the realised difference to the bound was 0.9934, so the constants are first-order tight.

Corollary 1 (Bounded trajectory functionals: why success rates are outside the theorem)**.** *Let f_θ map length-H trajectories to $[0,1]$. Then*

$$|\mathbb{E}_\theta^\pi f_\theta - \mathbb{E}_{\theta'}^\pi f_{\theta'}| \le \sup_\tau |f_\theta(\tau) - f_{\theta'}(\tau)| + H \sup_{(s,a)\in\mathcal{K}} \mathrm{TV}(P_\theta, P_{\theta'}),$$

$$|\mathbb{E}_\theta^\pi f - \mathbb{E}_\theta^{\pi'} f| \le H \sup_s \mathrm{TV}\big(\pi(\cdot \mid s), \pi'(\cdot \mid s)\big).$$

Both follow from centering with span one and the H-kernel telescoping of Appendix A.1. For a goal-conditioned success indicator, the first term can equal one because a trajectory can succeed for one goal and fail for another; for distinct deterministic policies, the policy-TV bound can be the trivial value H. Thus no generally non-trivial success-rate form of Theorem 1 follows. We therefore bound the dense Meta-World return, with $\Delta_r = 10$, and report success descriptively.

B Proofs for Theorem 2-A

The selection rule in full. *Selection as least squares.* The decoder is $\log p_\psi(x \mid s, a, z) = -\frac{1}{2}\|(x - f_\psi(s,a,z))/\sigma\|^2 - \log Z_\sigma$ with $f_i := f_\psi(\cdot,\cdot,\bar{z}_i)$ the mean of code i, which defines a deterministic model MDP $\hat{M}_i$. The context is $c^* = \{(s_j, a_j, x_j)\}_{j=1}^m$ with state-action pairs drawn from a context distribution ν^* on $\mathcal{K}$ (online, the exploration visitation) and outcomes from P_{θ^*}. All maximisers and minimisers below use a fixed deterministic tie-break. The rule scores every training code by $S_i := \sum_{j=1}^m \log p_\psi(x_j \mid s_j, a_j, \bar{z}_i)$, keeps $\hat{\imath} \in \arg\max_{i\in[n]} S_i$, and acts with $\pi_{\bar{z}_{\hat{\imath}}}^{\mathrm{ev}}$. Since the width is fixed, S_i is, up to a code-independent constant, minus half the sum of squared σ-scaled

residuals of code i: likelihood selection is least-squares selection of the training model that best predicts the context, with population counterpart

$$R_i := \mathbb{E}_{(s,a)\sim\nu^*}\|(\bar{x}^*(s,a) - f_i(s,a))/\sigma\|^2, \qquad i_{\mathrm{L}} \in \arg\min_i R_i, \qquad \Delta_i := \tfrac{1}{2}(R_i - R_{i_{\mathrm{L}}}) \geq 0, \tag{9}$$

where $\bar{x}^*$ is the true conditional mean. Here i_{L} is the likelihood-best code and Δ_i the per-transition margin of code i, which we call the *excitation gap* as it is the quantity that exploration has to make positive.

Three slacks. The selection theorem carries three slacks, stated as Assumptions (D1) to (D3) in Appendix B. Write $d_{\mathcal{K}}(\theta^*, \hat{M}_i)$ for the pointwise realised-reward Wasserstein pseudometric equation 6, specialised to the task and the model MDP of code i. The *decoder error* is $\delta := \max_i d_{\mathcal{K}}(\hat{M}_i, \theta_i)$, which accounts for states the decoder's training data never covered. The *sub-Gaussian parameter* σ_ℓ^2 describes the per-transition log-likelihood ratios. The *coverage slack* $(C_{\mathcal{K}}, \kappa)$ appears in $d_{\mathcal{K}}(\theta^*, \hat{M}_i) \leq C_{\mathcal{K}}\, d_{\nu^*}(\theta^*, \hat{M}_i) + \kappa$, where d_{ν^*} averages the pointwise metric integrand under the context distribution. With the curated-window decoder, scores on excited contexts were uninformative; training the decoder on all tiers improves coverage of the visited states.

Appendix C makes this quantitative in two cases that we can prove completely. For quadratic goal rewards (*bracketing excitation*), a positive-definite position covariance selects a code whose goal is within an explicit geometry-and-conditioning factor of the nearest training goal. One sufficient way to make the between-episode covariance positive definite is to use $K \geq d_{\mathrm{goal}} + 1$ episodes whose mean positions affinely span $\mathbb{R}^{d_{\mathrm{goal}}}$ (Proposition B.1, Corollary 3). For one-dimensional symmetric rewards (*monotone excitation*), exploration that stays on one side of all candidate speeds orders the codes exactly by the metric, while a single intermediate speed leaves its mirror image indistinguishable (Proposition B.2). For Meta-World's shaped rewards, we prove identifiability but not a quantitative gap (Appendix C.4).

B.1 Setup and assumptions

Throughout this appendix the simulators are deterministic, $P_\theta(\cdot \mid s, a) = \delta_{f_\theta(s,a)}$, as in every environment we use, and we take $\rho_{\mathcal{S}}(s, s') = \|s - s'\|_2$. Write $\Delta s_\theta(s,a) := f_\theta(s,a) - s$ and $r_\theta(s,a) := r_\theta(s, a, f_\theta(s,a))$ for the realised transition and reward, and $x^*(s,a) := (\Delta s_{\theta^*}(s,a), r_{\theta^*}(s,a))$. The decoder model has transition map $\hat{f}_i(s,a) := s + f_i^{\Delta s}(s,a)$ and realised reward $f_i^r(s,a)$; assume $\hat{f}_i(s,a) \in \mathcal{S}$ on $\mathcal{K}$. Thus the pointwise metric equation 6 between a task and the mean model $\hat{M}_i$ of code i has the integrand

$$\begin{aligned} \iota_i(s,a) &:= |r_{\theta^*}(s,a) - f_i^r(s,a)| + c_P\|\Delta s_{\theta^*}(s,a) - f_i^{\Delta s}(s,a)\|, \\ d_{\mathcal{K}}(\theta^*, \hat{M}_i) &:= \sup_{(s,a)\in\mathcal{K}} \iota_i, \qquad d_\nu(\theta^*, \hat{M}_i) := \mathbb{E}_\nu \iota_i, \end{aligned} \tag{10}$$

with one fixed transition coefficient c_P satisfying the common-coefficient conditions of the realised-reward form in Remark 2 for all training-code evaluation policies and both near-optimal comparison directions. Because c_P is fixed, $d_{\mathcal{K}}$ is a pseudometric and satisfies the triangle inequality across tasks and models; under the common-coefficient conditions it also supplies the return bounds in equation 6. The same integrand with θ_i in place of θ^* defines $d_{\mathcal{K}}(\hat{M}_i, \theta_i)$. We restrict this theorem to deterministic simulators; extending the likelihood argument to stochastic kernels would require controlling distributional, rather than merely conditional-mean, error. The selection objects are those of Section 5: the homoscedastic Gaussian decoder with per-coordinate widths $\sigma = (\sigma^{\Delta s}, \sigma_r)$, the scores S_i of Section 5, and R_i, i_{L}, Δ_i in equation 9.

Assumption D0 (Context sampling)**.** The context pairs (s_j, a_j), $j \leq m$, are i.i.d. from a distribution ν^* supported in $\mathcal{K}$, $x_j = x^*(s_j, a_j)$, and the Gaussian log scores are integrable. This is an idealised sampling assumption; the rollout contexts used in the experiments are dependent trajectories.

Assumption D1 (Decoder error on the reachable set)**.** $d_{\mathcal{K}}(\hat{M}_i, \theta_i) \leq \delta$ for every training code i.

Assumption D2 (Sub-Gaussian log-likelihood ratios)**.** For every $i \neq i_{\mathrm{L}}$ the per-transition ratio $\ell_j^{(i)} := \log p_\psi(x_j \mid s_j, a_j, \bar{z}_{i_{\mathrm{L}}}) - \log p_\psi(x_j \mid s_j, a_j, \bar{z}_i)$ is σ_ℓ^2-sub-Gaussian about its mean.

Assumption D3 (Coverage of the reachable set by the context)**.** There are $C_{\mathcal{K}} \geq 1$ and $\kappa \geq 0$ with $d_{\mathcal{K}}(\theta^*, \hat{M}_i) \leq C_{\mathcal{K}}\, d_{\nu^*}(\theta^*, \hat{M}_i) + \kappa$ for every i.

Assumption D1 is the decoder's own error as a model of the task it was fitted to. But critically, this error is measured over the whole reachable set, not just the decoder's own training support. For example, a decoder trained only on curated high-return data can have a small validation error on its own support, yet a large error on the low-reward states that exploration visits. δ charges for both kinds of error. Assumption D2 holds for the Gaussian decoder if the outcomes and decoder means are uniformly bounded on $\mathcal{K}$ and every fixed width is positive, because then

$$|\ell_j^{(i)}| \le L_\ell := \sum_k \frac{\|f_{i_L,k} - f_{i,k}\|_\infty \big(\|x_k\|_\infty + \|f_k\|_\infty\big)}{\sigma_k^2}, \tag{11}$$

where k ranges over the coordinates of x and the norms are suprema over $\mathcal{K}$ and over codes; a variable in $[-L_\ell, L_\ell]$ is sub-Gaussian with $\sigma_\ell^2 = L_\ell^2$. Assumption D3 is a separate uniform-coverage condition: it requires the average residual under the context distribution to control the worst residual on $\mathcal{K}$. Full topological support alone does not imply $\kappa = 0$; that conclusion additionally needs a finite comparison constant, as on a finite reachable set whose atoms have probability bounded away from zero. Neither δ nor κ is measured directly; the two knobs we turn experimentally, the decoder's training data and the exploration distribution, are intended to improve decoder accuracy on visited states and the empirical separation, but do not certify the global slacks.

B.2 Three lemmas

Lemma A (Population objective). *Under Assumption D0, $S_i/m \to -D_i$ almost surely as $m \to \infty$, where $D_i := \mathbb{E}_{\nu^*}\mathbb{E}_{P_{\theta^*}}[-\log p_\psi(x \mid s, a, \bar{z}_i)]$, and*

$$D_i = \tfrac{1}{2}R_i + \tfrac{1}{2}\mathbb{E}_{\nu^*}\mathbb{E}_{P_{\theta^*}}\|(x - \bar{x}^*)/\sigma\|^2 + \log Z_\sigma, \tag{12}$$

so that $D_i - D_{i'} = \frac{1}{2}(R_i - R_{i'})$ and $\arg\min_i D_i = \arg\min_i R_i$; in particular, their fixed-tie minimiser is i_L.

Proof. The summands of S_i are i.i.d. and integrable by Assumption D0, so the strong law gives the limit. Expanding $\|(x - f_i)/\sigma\|^2 = \|(x - \bar{x}^*)/\sigma\|^2 + 2\langle (x - \bar{x}^*)/\sigma, (\bar{x}^* - f_i)/\sigma\rangle + \|(\bar{x}^* - f_i)/\sigma\|^2$ and taking the conditional expectation given (s, a) kills the cross term because $\mathbb{E}[x - \bar{x}^* \mid s, a] = 0$; the middle term of equation 12 does not depend on i and vanishes for deterministic simulators. □

Lemma B (Finite-sample selection). *Under Assumptions D0 and D2, for every $i \neq i_L$, $\Pr(S_i \ge S_{i_L}) \le \exp(-m\Delta_i^2/(2\sigma_\ell^2))$, hence*

$$\Pr(\hat{\imath} \neq i_L) \le \sum_{i \neq i_L} \exp\Big(-\frac{m\Delta_i^2}{2\sigma_\ell^2}\Big) \le (n-1)\exp\Big(-\frac{m\Delta_{\min}^2}{2\sigma_\ell^2}\Big), \qquad \Delta_{\min} := \min_{i \neq i_L} \Delta_i. \tag{13}$$

Moreover, with $w := \mathrm{softmax}(S)$, on the event $\{\hat{\imath} = i_L\}$ we have $1 - w_{i_L} \le \sum_{i \neq i_L} \exp(-(S_{i_L} - S_i))$.

Proof. $S_{i_L} - S_i = \sum_j \ell_j^{(i)}$ with $\mathbb{E}\ell_j^{(i)} = D_i - D_{i_L} = \Delta_i$ by Lemma A. The event $\{S_i \ge S_{i_L}\}$ is $\{\sum_j (\ell_j^{(i)} - \Delta_i) \le -m\Delta_i\}$, and Hoeffding's inequality for i.i.d. σ_ℓ^2-sub-Gaussian variables (Boucheron et al., 2013) bounds its probability by $\exp(-(m\Delta_i)^2/(2m\sigma_\ell^2))$. The union bound over $i \neq i_L$ gives equation 13. For the softmax diagnostic, $1 - w_{i_L} = \sum_{i \neq i_L} e^{S_i} / \sum_k e^{S_k} \le \sum_{i \neq i_L} e^{S_i - S_{i_L}}$. □

Lemma C (From residuals to the task metric). *For every ν and i, $d_\nu(\theta^*, \hat{M}_i) \le c_\sigma \sqrt{R_i(\nu)}$ with $c_\sigma := \sigma_r + c_P \sigma_{\max}^{\Delta s}$, where $R_i(\nu)$ is equation 9 with ν in place of ν^* and $\sigma_{\max}^{\Delta s}$ is the largest next-state width.*

Proof. By Jensen,

$$\mathbb{E}_\nu |r_{\theta^*} - f_i^r| \le \sigma_r \Big(\mathbb{E}_\nu \frac{(r_{\theta^*} - f_i^r)^2}{\sigma_r^2}\Big)^{1/2},$$

$$\mathbb{E}_\nu \|\Delta s_{\theta^*} - f_i^{\Delta s}\| \le \sigma_{\max}^{\Delta s} \Big(\mathbb{E}_\nu \sum_k \frac{(\Delta s_{\theta^*,k} - f_{i,k}^{\Delta s})^2}{\sigma_k^2}\Big)^{1/2};$$

each root is at most $\sqrt{R_i(\nu)}$ because R_i is the sum of the two scaled squared residuals. □

B.3 Theorem 2-A and its proof

Theorem 2-A (restated). *For a deterministic simulator, under the setting of Theorem 1, the common-coefficient condition of the pointwise realised-reward form equation 6 for all training-code evaluation policies and both near-optimal comparison directions when dynamics differ, and Assumptions D0 to D3, with probability at least* $1-(n-1)\exp(-m\Delta^2_{\min}/(2\sigma_\ell^2))$ *the selected code is* i_{L} *and*

$$J^*(\theta^*) - J_{\theta^*}(\pi^{\mathrm{ev}}_{\bar{z}_{\hat{\imath}}}) \le \frac{2\big(C_{\mathcal{K}}\, c_\sigma \sqrt{\min_i R_i} + \kappa + \delta\big)}{1-\gamma} + \varepsilon_{i_{\mathrm{L}}}. \tag{14}$$

In the finite-horizon setting, suppose additionally that the fixed evaluation policies and the time-dependent ϵ-optimal policies in both comparison directions satisfy the realised-reward conditions of Lemma 1-H with a common $c_{P,H} \ge \max_{t<H} L_{V,t+1}$. Define $d_{\mathcal{K},H}$ and $d_{\nu^*,H}$ by equation 10 with $c_{P,H}$, and reimpose Assumptions D1 and D3 for these metrics with constants δ_H and $(C_{\mathcal{K},H}, \kappa_H)$. With $c_{\sigma,H} := \sigma_r + c_{P,H}\sigma^{\Delta s}_{\max}$, the same event obeys $J^*_H(\theta^*) - J^H_{\theta^*}(\pi^{\mathrm{ev}}_{\bar{z}_{\hat{\imath}}}) \le 2H(C_{\mathcal{K},H} c_{\sigma,H}\sqrt{\min_i R_i} + \kappa_H + \delta_H) + \varepsilon^H_{i_{\mathrm{L}}}$.

Proof. By Lemma B the event $\{\hat{\imath} = i_{\mathrm{L}}\}$ has the stated probability. On it, apply the pointwise realised-reward version of Theorem 1, justified by equation 6, with $i = i_{\mathrm{L}}$ and $\hat{z} = \bar{z}_{i_{\mathrm{L}}}$. Thus $C_{i_{\mathrm{L}}}(\bar{z}_{i_{\mathrm{L}}}) = 0$ and $J^*(\theta^*) - J_{\theta^*}(\pi^{\mathrm{ev}}_{\bar{z}_{i_{\mathrm{L}}}}) \le 2d_{\mathcal{K}}(\theta^*, \theta_{i_{\mathrm{L}}})/(1-\gamma) + \varepsilon_{i_{\mathrm{L}}}$. The pointwise metric with the fixed coefficient c_P satisfies the triangle inequality on $\mathcal{K}$, so $d_{\mathcal{K}}(\theta^*, \theta_{i_{\mathrm{L}}}) \le d_{\mathcal{K}}(\theta^*, \hat{M}_{i_{\mathrm{L}}}) + d_{\mathcal{K}}(\hat{M}_{i_{\mathrm{L}}}, \theta_{i_{\mathrm{L}}}) \le C_{\mathcal{K}} d_{\nu^*}(\theta^*, \hat{M}_{i_{\mathrm{L}}}) + \kappa + \delta$ by Assumptions D3 and D1, and $d_{\nu^*}(\theta^*, \hat{M}_{i_{\mathrm{L}}}) \le c_\sigma\sqrt{R_{i_{\mathrm{L}}}}$ by Lemma C; finally $R_{i_{\mathrm{L}}} = \min_i R_i$ by definition. For the finite-horizon statement, repeat the argument with the pointwise realised-reward bound of Lemma 1-H; its two task-transfer terms contribute $2Hd_{\mathcal{K},H}$. The finite-horizon versions of Assumptions D1 and D3, followed by Lemma C with coefficient $c_{P,H}$, give the displayed finite-horizon bound. The selection event is unchanged because it depends on the same scores. □

Corollary 2 (Expected gap). *Writing p_m for the right-hand side of equation 13 and $G_{i_{\mathrm{L}}}$ for the right-hand side of equation 14,* $\mathbb{E}[J^*(\theta^*) - J_{\theta^*}(\pi^{\mathrm{ev}}_{\bar{z}_{\hat{\imath}}})] \le G_{i_{\mathrm{L}}} + p_m \max_i\{2d_{\mathcal{K}}(\theta^*, \theta_i)/(1-\gamma) + \varepsilon_i\}$, *because on the misselection event the code is still a training code and the pointwise realised-reward version of Theorem 1 applies with $C_i = 0$.*

Remark 5 (Score with the active term of the metric). For reward-shift families the true dynamics are shared, so exact or code-independent transition predictions carry no task information and only add score variance. Code-dependent transition-model error can instead bias the population ranking. In either case the proof can be run deliberately on the reward-only pseudometric, with Assumptions D1 and D3 restricted to decoder reward error. Scoring with the reward coordinate alone then replaces R_i and σ_ℓ by their reward-only versions without losing a task-dependent transition term. For dynamics-shift families the next-state coordinates carry task signal and the full score is required.

Remark 6 (Why the decoder must be homoscedastic). If the decoder's width $\sigma_i(s,a)$ depends on the code, the score of code i contains $-\sum_k \log \sigma_{i,k}(s,a) - \|(x - f_i)/\sigma_i\|^2/2$. Under covariate shift, inflated predicted variance can reduce a large scaled residual enough to outweigh both mean accuracy and the log-width penalty, so the argmax is no longer a least-squares comparison of the means. We observed this failure mode on cheetah-vel, where a far training code won on out-of-support contexts. A fixed width removes this route, and the ranking then depends on residuals only, which is what Lemma A uses.

Remark 7 (The ceiling). Since $\bar{z}_{\hat{\imath}}$ is a training code, $J_{\theta^*}(\pi^{\mathrm{ev}}_{\bar{z}_{\hat{\imath}}}) \le \max_i J_{\theta^*}(\pi^{\mathrm{ev}}_{\bar{z}_i}) =: B$ for every selection rule. Its leave-one-out estimate is the B_{cv} of Section 4, so the gain of any code-set method over the encoder's code is at most $B - O$, estimated by the measured ceiling gap $B_{\mathrm{cv}} - O$. On cheetah-vel and ant-goal the encoder's code is already as good as the best training code and $B_{\mathrm{cv}} - O \approx 0$: no selection can help, and Theorem 2-A does not promise otherwise. In regime C, $B_{\mathrm{cv}} - O$ is large, so code-set selection has empirical headroom, subject to the theorem's assumptions.

Remark 8 (Trajectory data). Assumption D0 is a simplification: online contexts are trajectories, so consecutive pairs are dependent. A finite-sample analogue would require explicit mixing, martingale, or episodic-independence assumptions and a corresponding concentration argument. We do

not claim the displayed exponential probability for the trajectories in our experiments. The population quantities R_i and Δ_i can still be defined under the exploration visitation and explain what the selector scores asymptotically; the i.i.d. theorem isolates the additional sampling condition needed for its stated rate.

C EXCITATION: PROOFS FOR THEOREM 2-B

C.1 THEOREM 2-B AND ITS PROOF

Theorem 2-B (restated). *(i) Let ν_{e} be the state-action visitation of an exploration policy on θ^* and let the context be m transitions from $\nu_{\mathrm{e}} \otimes P_{\theta^*}$. Under all assumptions of Theorem 2-A, with $\nu^* = \nu_{\mathrm{e}}$, if $\Delta_{\min}(\nu_{\mathrm{e}}) := \min_{i \neq i_{\mathrm{L}}} \Delta_i(\nu_{\mathrm{e}}) > 0$, then the conclusions of Theorem 2-A hold. (ii) Fix m. Suppose two tasks θ_1^*, θ_2^* induce the same full law of the length-m context under the exploration procedure and have distinct unique return-best training codes $i_k := \arg\max_{i \in [n]} J_{\theta_k^*}(\pi_{\tilde{z}_i}^{\mathrm{ev}})$. Then every selection rule $\hat{\imath}$, measurable in the context and in independent auxiliary randomness, satisfies $\max_k \Pr_{\theta_k^*}(\hat{\imath} \neq i_k) \geq \frac{1}{2}$; for $|I_0|$ tasks with the same context law and pairwise distinct unique return-best training codes, $\max_{k \in I_0} \Pr_{\theta_k^*}(\hat{\imath} \neq i_k) \geq 1 - 1/|I_0|$.*

Proof. (i) is Theorem 2-A with the exploration visitation as the context distribution; all of that theorem's assumptions, including (D0)–(D3), remain in force. (ii) The law of the context is the same under the two tasks, and $\hat{\imath}$ is a function of the context and of randomness independent of the task, so the law of $\hat{\imath}$ is the same under both. Because $i_1 \neq i_2$, the events $\{\hat{\imath} \neq i_1\}$ and $\{\hat{\imath} \neq i_2\}$ cover the whole space, so $\Pr(\hat{\imath} \neq i_1) + \Pr(\hat{\imath} \neq i_2) \geq 1$ under the common law, and the larger of the two is at least one half. For $|I_0|$ tasks, $\sum_{k \in I_0} \Pr(\hat{\imath} = i_k) \leq 1$ since the i_k are distinct, hence $\sum_k \Pr(\hat{\imath} \neq i_k) \geq |I_0| - 1$ and the maximum is at least $1 - 1/|I_0|$. □

For reward-shift families with shared dynamics and a reward-blind exploration policy, the visitation ν_{e} is the same for every task, and the hypothesis of (ii) reduces to the level-set condition $r_{\theta_1^*} = r_{\theta_2^*}$ ν_{e}-almost surely: the exploration visited only places where the two tasks pay the same. The rest of this appendix quantifies when exploration avoids this.

C.2 BRACKETING EXCITATION FOR QUADRATIC GOAL REWARDS

Consider goals $g \in G \subset \mathbb{R}^d$, a position feature $h : \mathcal{S} \to \mathbb{R}^d$, shared dynamics, and rewards $r_g(s,a) = -\|h(s) - g\|^2 + b(s,a)$ with a goal-independent shaping term b. Training goals are $g_1, \ldots, g_n$, the target goal is g^*, the reward decoder is exact, $f_i^r = r_{g_i}$, and the score is reward-only. Throughout this subsection, $S_i, R_i, \Delta_i, \ell^{(i)}$, and σ_ℓ denote their reward-only versions. Assume $\mathbb{E}_{\nu_{\mathrm{e}}}\|h\|^2 < \infty$. Write $\bar{h} := \mathbb{E}_{\nu_{\mathrm{e}}} h$, $\lambda_{\mathrm{e}} := \lambda_{\min}(\mathrm{Cov}_{\nu_{\mathrm{e}}}(h))$ for the excitation level, $\Lambda_{\mathrm{e}} := \lambda_{\max}(\mathrm{Cov}_{\nu_{\mathrm{e}}}(h)) + \sup_{c \in \mathrm{conv}(G \cup \{g^*\})} \|\bar{h} - c\|^2$, $\rho^* := \min_j \|g^* - g_j\|$ for the distance to the nearest training goal, and $\rho_{\mathcal{K}} := \sup\{\|h(s) - c\| : (s,a) \in \mathcal{K},\ c \in \mathrm{conv}(G \cup \{g^*\})\}$. Assume also that Λ_{e} and $\rho_{\mathcal{K}}$ are finite.

Proposition B.1 (Bracketing excitation). *(a) For every i, $\frac{4\lambda_{\mathrm{e}}}{\sigma_r^2}\|g^* - g_i\|^2 \leq R_i \leq \frac{4\Lambda_{\mathrm{e}}}{\sigma_r^2}\|g^* - g_i\|^2$. If $\lambda_{\mathrm{e}} > 0$, then (b) the likelihood-best code satisfies $\|g^* - g_{i_{\mathrm{L}}}\| \leq \sqrt{\Lambda_{\mathrm{e}}/\lambda_{\mathrm{e}}}\,\rho^*$; (c) $\Delta_i \geq \frac{2}{\sigma_r^2}\big(\lambda_{\mathrm{e}}\|g^* - g_i\|^2 - \Lambda_{\mathrm{e}}\rho^{*2}\big)$, which is positive for every code outside the band $B_{\mathrm{e}} := \{i : \|g^* - g_i\| \leq \sqrt{\Lambda_{\mathrm{e}}/\lambda_{\mathrm{e}}}\,\rho^*\}$; and (d), under Assumptions D0 and D2, with probability at least $1 - \sum_{i \notin B_{\mathrm{e}}} \exp(-m\Delta_i^2/(2\sigma_\ell^2))$ the selected code lies in B_{e} and*

$$J^*(\theta^*) - J_{\theta^*}(\pi_{\tilde{z}_{\hat{\imath}}}^{\mathrm{ev}}) \leq \frac{4\rho_{\mathcal{K}}\sqrt{\Lambda_{\mathrm{e}}/\lambda_{\mathrm{e}}}\,\rho^*}{1-\gamma} + \max_{i \in B_{\mathrm{e}}} \varepsilon_i. \tag{15}$$

Proof. (a) Expanding the squares, $r_{g^*} - r_{g_i} = \|h - g_i\|^2 - \|h - g^*\|^2 = 2(h - c_i)^\top (g^* - g_i)$ with $c_i := (g^* + g_i)/2$, which is affine in the position. Hence $R_i = \mathbb{E}_{\nu_{\mathrm{e}}}(r_{g^*} - r_{g_i})^2/\sigma_r^2 = (4/\sigma_r^2)(g^* - g_i)^\top M(c_i)(g^* - g_i)$ with $M(c) := \mathbb{E}_{\nu_{\mathrm{e}}}[(h-c)(h-c)^\top] = \mathrm{Cov}_{\nu_{\mathrm{e}}}(h) + (\bar{h}-c)(\bar{h}-c)^\top$. The second

summand is positive semidefinite, so $\lambda_{\min}(M(c)) \geq \lambda_{\mathrm{e}}$, and $\lambda_{\max}(M(c)) \leq \lambda_{\max}(\mathrm{Cov}_{\nu_{\mathrm{e}}}(h)) + \|\bar{h} - c\|^2 \leq \Lambda_{\mathrm{e}}$ since $c_i \in \mathrm{conv}(G \cup \{g^*\})$. (b) Let i_{near} attain ρ^*. Then $4\lambda_{\mathrm{e}}\|g^* - g_{i_{\mathrm{L}}}\|^2/\sigma_r^2 \leq R_{i_{\mathrm{L}}} \leq R_{i_{\mathrm{near}}} \leq 4\Lambda_{\mathrm{e}}\rho^{*2}/\sigma_r^2$. (c) $\Delta_i = \frac{1}{2}(R_i - R_{i_{\mathrm{L}}}) \geq \frac{1}{2}(4\lambda_{\mathrm{e}}\|g^* - g_i\|^2 - 4\Lambda_{\mathrm{e}}\rho^{*2})/\sigma_r^2$ by (a) and the last inequality in (b). (d) By Lemma B, for each $i \notin B_{\mathrm{e}}$ the event $\{S_i \geq S_{i_{\mathrm{L}}}\}$ has probability at most $\exp(-m\Delta_i^2/(2\sigma_\ell^2))$. Outside the union of these events every code outside the band scores strictly below i_{L}, and $i_{\mathrm{L}} \in B_{\mathrm{e}}$ by (b), so the maximiser $\hat{\imath}$ lies in B_{e}. Theorem 1 with $i = \hat{\imath}$ and $\hat{z} = \bar{z}_{\hat{\imath}}$ gives a gap of at most $2d(\theta^*, \theta_{\hat{\imath}})/(1-\gamma) + \varepsilon_{\hat{\imath}}$, and $d(\theta^*, \theta_{\hat{\imath}}) = \sup_{(s,a)\in\mathcal{K}} |2(h - c_{\hat{\imath}})^\top (g^* - g_{\hat{\imath}})| \leq 2\rho_{\mathcal{K}}\|g^* - g_{\hat{\imath}}\| \leq 2\rho_{\mathcal{K}}\sqrt{\Lambda_{\mathrm{e}}/\lambda_{\mathrm{e}}}\,\rho^*$. □

The proposition says what bracketing buys. The smallest eigenvalue of the exploration's position covariance lower-bounds its worst-direction sensitivity to the goal. The ratio $\Lambda_{\mathrm{e}}/\lambda_{\mathrm{e}}$ combines covariance anisotropy with the offset of the mean position from the goal set, and the selected goal is within its square-root factor of the nearest training goal. When $\lambda_{\mathrm{e}} = 0$, parts (b)–(d) give no guarantee. In the special case where exploration is concentrated at one position h_0, part (ii) of Theorem 2-B applies to any two tasks on the sphere $\{g : \|h_0 - g\| = \|h_0 - g^*\|\}$ that require different unique return-best training codes: they give the same reward at the only visited position.

Corollary 3 ($K = d_{\mathrm{goal}} + 1$ exploration codes)**.** *Let the exploration consist of K episodes of equal length with codes $z_{\mathrm{e},1}, \ldots, z_{\mathrm{e},K}$, so that ν_{e} is the uniform mixture of the episode visitations, and let $\bar{h}_k$ be the mean position in episode k. Then $\mathrm{Cov}_{\nu_{\mathrm{e}}}(h) \succeq \frac{1}{K}\sum_k (\bar{h}_k - \bar{h})(\bar{h}_k - \bar{h})^\top$. The matrix on the right is positive definite if and only if the K episode means affinely span $\mathbb{R}^d$; this between-episode condition requires $K \geq d + 1$, and $K = d + 1$ affinely independent means suffice. Within-episode motion can add excitation, so $K \geq d + 1$ is sufficient for this construction but is not necessary for the full covariance to be positive definite.*

Proof. The law of total covariance gives $\mathrm{Cov}_{\nu_{\mathrm{e}}}(h) = \frac{1}{K}\sum_k \mathrm{Cov}_k(h) + \frac{1}{K}\sum_k (\bar{h}_k - \bar{h})(\bar{h}_k - \bar{h})^\top$, and the first sum is positive semidefinite. The second is positive definite iff no nonzero vector is orthogonal to every $\bar{h}_k - \bar{h}$, i.e. iff the means are affinely spanning. □

The condition is the persistence-of-excitation requirement of adaptive identification (Narendra & Annaswamy, 1987) transposed to code space: a least-squares selector separates parameters when the relevant regressor second-moment matrix is uniformly positive definite. One episode can already provide full-rank excitation through within-episode motion; concentration at a single position is the canonical failure case. The corollary motivates exploring with several training codes and treating farthest-point sampling in code space as a heuristic: it may spread episode-mean positions when the code-to-behaviour map is sufficiently regular, but code-space separation alone gives no such guarantee. The condition is stated in terms of the positions actually visited, which is what should be logged. The exploration set must also stay inside the decoder's support, since a code that drives the system into states the decoder never saw raises δ on exactly the states that enter the score, the failure we observed with extreme-velocity codes on cheetah-vel.

C.3 Monotone excitation for one-dimensional symmetric rewards

Consider $r_{v^*}(s,a) = -|v(s,a) - v^*| + b(s,a)$ with a scalar speed feature v, candidate speeds $v_1, \ldots, v_n$, target speed v^*, exact reward decoder and reward-only score. Throughout this subsection we use the reward-only pseudometric of Remark 5, and $S_i, R_i, \Delta_i, \ell^{(i)}$, and σ_ℓ denote their reward-only versions. The reverse triangle inequality gives $d(\theta^*, \theta_i) \leq |v^* - v_i|$; equality holds when $\mathcal{K}$ contains a state-action pair whose speed lies outside the interval between v^* and v_i.

Proposition B.2 (Monotone excitation and two-sided level sets)**.** *(a) If $v(s,a) \leq \min(v^*, \min_i v_i)$ for ν_{e}-almost every (s,a) (or $\geq$ the maximum), then $r_{v^*} - r_{v_i} = v_i - v^*$ ν_{e}-almost surely, hence $R_i = (v_i - v^*)^2/\sigma_r^2$, the likelihood-best code is a metric-nearest code, and $\Delta_i = \big((v_i - v^*)^2 - \rho^{*2}\big)/(2\sigma_r^2)$ with $\rho^* = \min_j |v_j - v^*|$. In the reward-only specialization, Assumption D1 holds with $\delta = 0$ and Assumption D3 holds with $C_{\mathcal{K}} = 1$, $\kappa = 0$. If the nearest code is unique and Assumptions D0 and D2 hold, the reward-only form of Theorem 2-A gives, with probability at least $1 - (n-1)\exp(-m\Delta_{\min}^2/(2\sigma_\ell^2))$, $J^*(\theta^*) - J_{\theta^*}(\pi_{\bar{z}_{\hat{\imath}}}^{\mathrm{ev}}) \leq 2\rho^*/(1-\gamma) + \varepsilon_{i_{\mathrm{L}}}$. (b) If instead ν_{e} is concentrated on a single speed v_0 strictly between two task speeds, then $r_{v^*}(v_0) = r_{v'}(v_0)$ for the mirror speed $v' = 2v_0 - v^*$. Under shared dynamics and reward-blind exploration this makes*

the full context laws coincide for every m. Thus, whenever the mirror speed is itself a task with a different unique return-best training code, Theorem 2-B(ii) gives an error probability of at least one half for one of the tasks, for every m. (c) If ν_e visits two distinct speeds $v_0 \neq v_1$, the map $v^* \mapsto (r_{v^*}(v_0), r_{v^*}(v_1))$ is injective, so reward-based identifiability holds in principle.

Proof. (a) On the region $v \leq \min(v^*, v_i)$ both absolute values open with the same sign, $r_{v^*} - r_{v_i} = (v - v^*) - (v - v_i) = v_i - v^*$, a constant. Then R_i, i_L and Δ_i follow from equation 9. The reward-only integrand equals $|v_i - v^*|$ ν_e-almost surely, so $d_{\nu_e} = |v_i - v^*|$. The reverse-triangle upper bound and the presence in $\mathcal{K}$ of the one-sided visited points give $d_{\mathcal{K}} = |v_i - v^*|$ as well. Exact reward decoding gives $\delta = 0$, and Assumption D3 follows with $C_{\mathcal{K}} = 1$, $\kappa = 0$; the displayed bound is equation 14 with $c_\sigma \sqrt{R_{i_L}} = \rho^*$. (b) $|v_0 - v^*| = |v_0 - v'|$ for $v' = 2v_0 - v^*$. (c) If $v' \neq v^*$ satisfied $|v_0 - v'| = |v_0 - v^*|$ and $|v_1 - v'| = |v_1 - v^*|$, then $v' = 2v_0 - v^* = 2v_1 - v^*$, forcing $v_0 = v_1$. □

Part (a) is the cleanest case in the paper: an exploration that stays on one side of every candidate makes the reward affine in the task parameter over the visited region, the score becomes an exact plug-in of the metric, and there is no conditioning loss. For velocity tasks the prior code $z = 0$ runs slowly, below every training speed, so it is the natural outside explorer, and this is the exploration we use there. Part (b) is the failure mode represented by exploration concentrated at an intermediate speed: the level set of one visited speed is a pair of points, and the training code on the wrong branch fits as well as the right one. Part (c) says that two visited speeds identify the task in principle. The resulting score separation depends on how much context is collected at the second speed, and decoder misspecification on states reached by fast codes can distort that separation. This is qualitatively consistent with the confident wrong-branch selections we saw when fast-code episodes dominated the summed likelihood, but we do not have a quantitative statement for that regime.

C.4 What is and is not proved for shaped goal rewards

Meta-World rewards have the form $r_g = \phi(\|h(s) - g\|) + (\text{goal-independent terms})$ with a continuous, strictly monotone ϕ over the relevant range, and in the contact tasks additional grasp and object terms. For this simplified form we can prove identifiability under the trilateration condition: if the support of the pushforward distribution $h_{\#}\nu_e$ contains $d + 1$ affinely independent positions $h_0, \ldots, h_d$ and ϕ is injective on the realised distances, then the rewards at those positions determine g, because ϕ^{-1} recovers the distances $\|h_k - g\|$ and subtracting the squared-distance equations for $k = 1, \ldots, d$ from that for $k = 0$ gives a linear system $2(h_k - h_0)^\top g = \|h_k\|^2 - \|h_0\|^2 - \|h_k - g\|^2 + \|h_0 - g\|^2$ whose coefficient rows are linearly independent. Moreover, $h \mapsto \phi(\|h - g^*\|) - \phi(\|h - g'\|)$ is continuous, so a zero squared residual $h_{\#}\nu_e$-almost surely is zero on the support. Hence $\mathbb{E}_{\nu_e}(r_{g^*} - r_{g'})^2 = 0$ implies $g' = g^*$ under these conditions. This rules out a distinct task with zero residual, but it does not rule out equal positive residuals—and hence score ties—between two candidate goals. What we do not have is a quantitative lower bound on $\mathbb{E}_{\nu_e}(r_{g^*} - r_{g'})^2$ in terms of $\|g^* - g'\|$ for a saturating or nearly flat ϕ; the quadratic case of Proposition B.1 is the one we prove completely. For the contact tasks the reward is a composite of object-goal and grasp terms and the decoder's reward discriminability is empirically weaker, which places them outside the scope of Proposition B.1.

C.5 Reading the theorems against the experiments

The excitation gap Δ_i is a design quantity, not a task property. Through the population residuals, the exploration distribution can change i_L, $\min_i R_i$, every Δ_i, and the indexed offline gap ε_{i_L}; it can also change the likelihood-ratio tail parameter and both coverage constants $(C_{\mathcal{K}}, \kappa)$. The experiments vary practical levers related to these quantities rather than measuring the theorem's global constants. Changing the decoder's training data from the curated high-return window to all snapshot tiers can reduce decoder error on the low-reward states that exploration visits; the policy and encoder are untouched. Changing the exploration from the prior code to K farthest-point training codes can enlarge behavioural separation, but only the realised visitation—not code-space spread alone—determines $\Delta_{\min}$, λ_e, and the coverage condition. The in-distribution adaptation of Wang et al. (2023) samples task hypotheses from its belief, rolls out the corresponding conditioned policy, and is therefore aligned with the excitation principle; whether it satisfies the condition still depends

on $\Delta_{\min}(\nu_e) > 0$. Its return-based acceptance rule encourages in-support contexts, which may reduce analogues of κ and decoder error, but does not enforce either by construction and can accept nothing. The likelihood head needs no successful episode and works from reward variation alone. Accordingly, the hybrid uses the encoder when it has an accepted successful episode and likelihood otherwise; when VeriCode reaches this branch, it records both candidates and normally verifies them (Section 5).

D Proof of Proposition 3 and the repeatability measurement

Proof. Let $X_1, \dots, X_v$ be the success indicators for z_1 and $Y_1, \dots, Y_v$ those for z_2, with all $2v$ variables independent and with means p_1 and p_2. The rule commits to z_2 only on $\{\bar{Y} > \bar{X}\}$. Apply Hoeffding's inequality to the weighted sum $\bar{X} - \bar{Y}$; its $2v$ independent summands each have range length $1/v$, so

$$\Pr(\bar{Y} > \bar{X}) \le \Pr\big((\bar{X} - \bar{Y}) - (p_1 - p_2) \le -(p_1 - p_2)\big) \le \exp\big(-v(p_1 - p_2)^2\big).$$

If success is a deterministic function of the code, $p_1, p_2 \in \{0, 1\}$ and every sample mean equals its expectation, so the committed code has success $\max\{p_1, p_2\}$ with probability one for every $v \ge 1$. VeriCode's tie-break toward $\hat{z}_{\text{enc}}$ and its return comparison when both candidates fail only decide among candidates with equal success. □

Reused outcomes and agreement. If the likelihood-selected training code was already used during excitation, its recorded outcome is reused; under the measured repeatability of this harness, that outcome equals a new verification outcome. This reuse is not covered by the fresh-sample statement of Proposition 3 in general. If the selected code is also the training code nearest the encoder output, the agreement shortcut commits it without verification. This is the one branch in which an unsuccessful code can be committed without a deployment-time test, and it accounts for 10 percent of trials.

Proposition 3-R (Return-verified commitment)**.** *Let $z_1, \dots, z_K$ be candidate codes. For each k, let $G_{k,1}, \dots, G_{k,v}$ be evaluation returns with common mean J_k and values in an interval of width Δ_G, and assume that all Kv returns are mutually independent. Let $k^* = \arg\max_k J_k$ and $\Delta_k := J_{k^*} - J_k$. If the agent commits to an index $\hat{k}$ having the largest sample-mean return and $J_{\text{commit}} := J_{\hat{k}}$ denotes the population mean return of the committed code, then $\Pr(\hat{k} = k) \le \exp(-v\Delta_k^2/(2\Delta_G^2))$ for every $k \ne k^*$, and $\mathbb{E}\, J_{\text{commit}} \ge J_{k^*} - \sum_{k \ne k^*} \Delta_k e^{-v\Delta_k^2/(2\Delta_G^2)} \ge J_{k^*} - (K-1)\Delta_G/\sqrt{ev}$.*

Proof. Pair the episodes: $D_j := G_{k,j} - G_{k^*,j}$, $j \le v$, are independent with range $2\Delta_G$ and mean $-\Delta_k$, so Hoeffding gives $\Pr(\bar{D} \ge 0) \le \exp(-2v\Delta_k^2/(2\Delta_G)^2)$. The expectation bound uses $\mathbb{E}J_{\text{commit}} = J_{k^*} - \sum_k \Delta_k \Pr(\text{commit } k)$ and $\sup_{x \ge 0} x\, e^{-vx^2/(2\Delta_G^2)} = \Delta_G/\sqrt{ev}$. □

The proposition assumes a fixed candidate set and fresh independent samples. VeriCode reuses returns from its predetermined exploration codes and forms other candidates from the resulting context, so the displayed finite-sample guarantee does not directly cover the entire adaptive rule. Reuse is exact for deterministic returns; otherwise it is an empirical budget-saving heuristic, and fresh verification is needed for the stated guarantee.

Repeatability measurement. The diagnosis of Section 4 rolls out every one of the twenty training codes three times on every OOD task of every checkpoint. Over the six Meta-World families this gives more than 15,000 task-code pairs, and in every pair the three rollouts agree on success: the number of pairs with exactly one or two successes out of three is zero in every family. The evaluation actor is deterministic and the environment reset is seeded, so success given a code is a deterministic function of the task in this harness, which is the case of the proposition in which $v = 1$ is exact. We report it as a property of the harness rather than of the method: a stochastic reset would move the guarantee to the exponential rate, which is weak when the two candidates have close success probabilities.

E EXCLUDED ENVIRONMENTS: WHEN THE ID GAP DOMINATES

We ran the decomposition on twelve environments and excluded two by the rule stated in Section 4. Table 3 shows why: on walker-mass and walker-friction the ID gap is the largest term for at least two of the three algorithms, so the frozen policy fails on the nearest training task itself and no test-time inference rule can recover the OOD gap through that task. The same two families have the largest best-training-code headroom of the grid ($B_{\mathrm{cv}} - O$ of 66 to 140 return), which is why Section 6 evaluates VeriCode on them: the ID gap says the nearest training task is not solved, not that no training code helps.

Table 3: The two walker families, excluded from the diagnosis grid because the in-distribution gap dominates (OOD split; normalised terms as in Table 15). Exclusion rule: $\varepsilon/(E-L) > 0.5$ for at least two of the three baselines. Bold marks the exclusion statistic. For comparison, the largest ε share on the retained grid is hopper-mass ER-TRL (0.89, a single baseline; the rule requires two of three, so the family stays), and every Meta-World reach and reach-wall cell has $\varepsilon = 0$.

env	algo	seeds	$E-L$	Δ_{extrap}	ε	Δ_{latent}	Δ_{ctx}	$\varepsilon/(E-L)$	dominant
walker-mass	CSRO	6	0.57	−0.06	0.61	−0.02	0.03	**1.07**	ε
	ER-TRL	6	0.46	0.04	0.42	0.09	−0.09	**0.92**	ε
	UNICORN	6	0.63	−0.09	0.47	0.14	0.11	**0.75**	ε
walker-friction	CSRO	6	0.48	0.18	0.36	−0.09	0.03	**0.75**	ε
	ER-TRL	5	0.36	0.07	0.48	−0.20	0.00	**1.36**	ε
	UNICORN	6	0.56	−0.01	0.46	0.02	0.08	**0.82**	ε

F MUJOCO ONLINE ROWS: THE SELECTION HEAD ALONE, AND THE RULE WITHOUT THE PRIOR CANDIDATE

The rows below predate the return-comparison rule of Section 5 and report the selection head alone; the main MuJoCo result is Table 14. One ablation matters for the design: with five random-training-code exploration episodes and the prior code absent from the candidate set, the return-based rule wins all twelve dynamics-family cells and the three ant-goal cells but loses to the framework's channel on cheetah-vel by 28, 28 and 82 return (CSRO, ER-TRL, UNICORN). The pattern is consistent with the two-sided ambiguity of Proposition B.2: the context visited speeds on both sides of the candidate set. Adding the prior code's re-inferred candidate turns the three losses into gains of 22, 22 and 3 return.

G MUJOCO ONLINE ROWS: A NO-HARM CHECK IN REGIME A

The main MuJoCo rule is reported in Table 14; this section records two earlier selection-head variants used to choose its candidate set. On cheetah-vel the non-prior channel already matches the oracle offline-context return for CSRO and ER-TRL and is within about 20 return of it for UNICORN, so there is little inference gap to close; on cheetah-vel and ant-goal the ceiling $B_{\mathrm{cv}} - O$ is within noise, and on the hopper families it is 23 to 78 return that neither precursor variant below recovers. Their reversal across reward geometries motivates combining the prior and diverse-code candidates in the main rule.

The pattern in Table 4 illustrates the role of exploration geometry, and it reverses between the two task families. On the one-dimensional symmetric velocity reward, prior-code excitation matches the framework's online channel to within 6 return, but trails the non-prior channel by 22 to 41. Bracketing loses a further 33 to 62, consistent with the two-sided reward level sets of Proposition B.2. On the two-dimensional ant-goal task the ordering flips: diverse-code exploration beats non-prior by 49 and 82 for CSRO and UNICORN and by 10 for ER-TRL, while prior-code excitation, which barely moves the ant, is 15 to 60 below non-prior. This agrees qualitatively with Proposition B.1, but farthest-point codes do not by themselves certify its position-covariance condition. The hopper rows go both ways: hopper-friction ER-TRL loses 68 return and UNICORN gains 32 against non-prior, both beyond two s.e.m., and the remaining four hopper cells lie within two s.e.m. of zero. No single exploration choice is uniformly best.

Table 4: MuJoCo online no-harm check on the OOD split: undiscounted returns over six seeds. Columns: the framework's online channel ($z = 0$ exploration, encoder re-inference); the non-prior channel; oracle offline context; excite-then-select with the prior code $z = 0$ as the only exploration code and the all-tier decoder (v9, the monotone-excitation variant); the encoder re-inferred from the same v9 context; the fraction of OOD tasks on which v9 picked the parameter-nearest code; and the bracketing variant that adds farthest-point training codes to $z = 0$, re-explores once with the code selected on that context, and scores on the union of all episodes (v7). Δ is v9 minus non-prior with seed s.e.m.; ** marks $|\Delta| > 2$ s.e.m. and * marks $|\Delta| > 1$ s.e.m.

env	algo	fw. online	non-prior	offline ctx	v9 select (z=0)	v9 encoder	v9 picked nearest	v7 bracketing	Δ (v9 − nonprior)
cheetah-vel	CSRO	−140	−97	−99	−134	−140	0.61	−167	−37±16**
	ER-TRL	−140	−102	−100	−143	−138	0.54	−205	−41±8**
	UNICORN	−131	−115	−95	−137	−134	0.53	−192	−22±7**
ant-goal	CSRO	−361	−312	−223	−372	−363	0.13	−262	−60±26**
	ER-TRL	−346	−325	−262	−370	−357	0.15	−315	−45±13**
	UNICORN	−376	−362	−266	−377	−395	0.16	−281	−15±14*
hopper-mass	CSRO	521	553	553	512	477	0.39	510	−41±27*
	ER-TRL	529	479	502	451	474	0.43	453	−29±26*
	UNICORN	384	452	497	387	323	0.30	430	−65±76
hopper-friction	CSRO	415	401	449	437	404	0.88	431	+36±22*
	ER-TRL	360	403	396	336	342	0.67	344	−68±34**
	UNICORN	433	396	438	427	406	0.21	386	+32±15**

H EMPIRICAL METRIC DIAGNOSTICS AND EXPERT REGRET

The metric diagnostic. Lemma 1 supplies a linear upper envelope in the task pseudometric; it does not predict proportionality or correlation. The fits reported here use an earlier evaluation campaign with the same checkpoints and splits, which is why their points and slopes differ slightly from the six-repeat wave behind Table 15; the qualitative ordering of the families is the same. The least-squares fit over OOD tasks, seeds and algorithms has Pearson $r = 0.86$ on cheetah-vel, $r = 0.96$ on reach and reach-wall on the normalised scale, $r = 0.45$ on ant-goal, and $r = 0.16$ and -0.16 on hopper-mass and hopper-friction (Figure 4 and Table 5). These are empirical associations with parameter-distance proxies. A nominal envelope formed by substituting expert returns and estimated distances covers 92 to 100 percent of the cheetah-vel and ant-goal points (per-algorithm fractions 0.97, 0.95, 1.00 and 0.92, 0.95, 0.97). Because that calculation substitutes experts for optima and proxies for the theorem's global metric and constants, it is neither a verification nor a falsification of Theorem 1.

Table 5: Least-squares fit of the realised expert-reference extrapolation term Δ_{extrap} against task-parameter distance proxies across OOD tasks, with algorithms pooled (one point per task, seed, and algorithm; both OOD splits). Columns give the sample size; slope, through-origin slope, intercept, Pearson r, and Spearman ρ of the raw fit; the through-origin slope and Pearson r after normalisation; and per-algorithm normalised Pearson correlations. The proxies are $|\Delta v|$ for cheetah-vel, $\|\Delta g\|_1$ for ant-goal, $|\Delta \log \mathrm{scale}|$ for hopper and walker, and goal displacement for Meta-World. Only the first two are analytic reward-distance bounds; the dynamics proxies are assessed empirically in Table 6.

		raw fit					normalised fit		per-algo r
env	n	slope	slope$_0$	intercept	r	ρ	slope$_0$	r	(normalised)
cheetah-vel	360	178.6	159.3	−13.5	0.86	0.88	0.35	0.86	0.90/0.81/0.87
ant-goal	360	164.9	152.1	−8.8	0.45	0.41	0.20	0.46	0.46/0.57/0.37
hopper-mass	360	195.4	458.5	76.6	0.16	0.20	0.77	0.18	0.34/0.20/0.03
hopper-friction	360	−191.9	243.8	126.8	−0.16	−0.14	0.51	−0.08	−0.03/−0.14/−0.08
reach	280	6.3	6.0	−0.1	0.84	0.81	3.53	0.96	0.96/0.96/0.96
reach-wall	240	7.1	5.8	−0.2	0.92	0.86	3.44	0.96	0.96/0.95/0.96
push	320	6.3	6.0	0.0	0.68	0.63	4.74	0.52	0.47/0.56/0.55
pick-place	300	4.2	3.5	−0.1	0.71	0.62	1.62	0.44	0.50/0.46/0.38
soccer	360	11.1	4.9	−0.8	0.59	0.59	4.41	0.37	0.45/0.45/0.28
plate-slide	300	7.9	6.8	−0.1	0.84	0.80	5.64	0.84	0.87/0.87/0.81
walker-mass (excluded)	360	−39.9	−138.0	−28.6	−0.02	−0.04	−0.23	−0.04	−0.08/−0.02/−0.02
walker-friction (excluded)	360	214.4	226.1	3.4	0.12	0.13	0.25	0.11	0.15/0.11/0.06

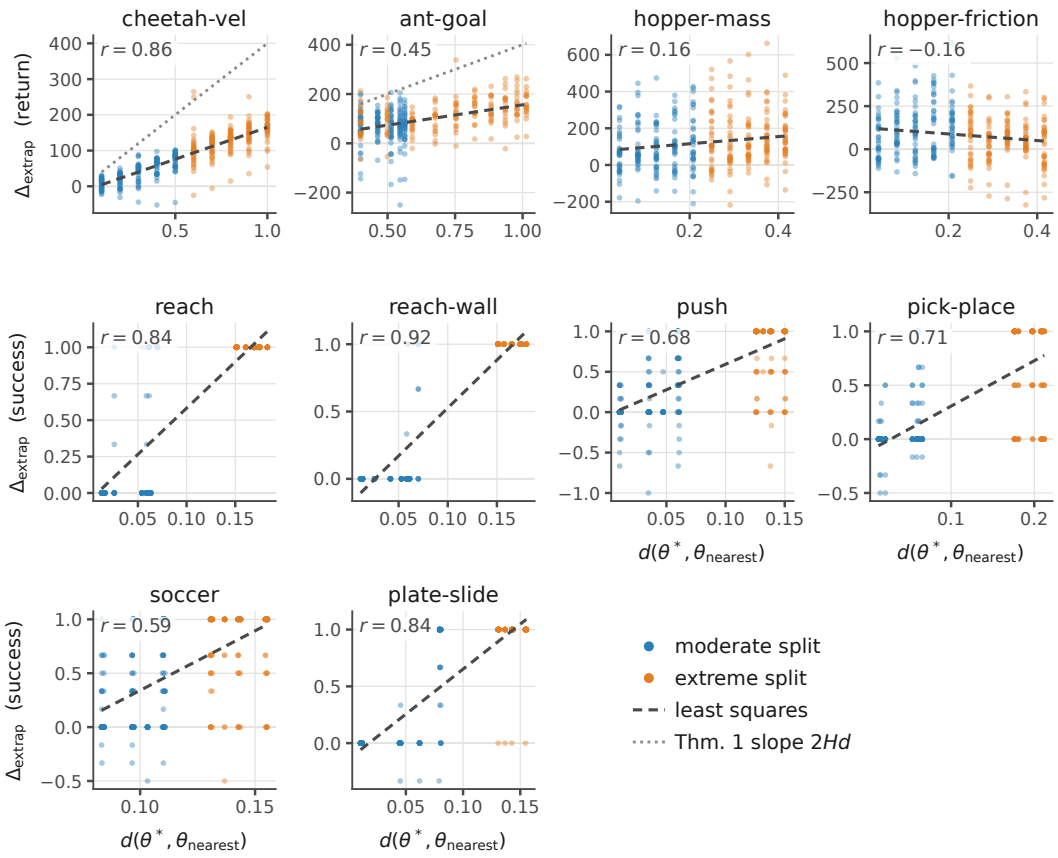


Figure 4: Empirical association between the realised expert-reference extrapolation term $\Delta_{\text{extrap}} = (E - N) - \varepsilon$ and a task-parameter distance proxy, one point per OOD task, seed and algorithm. Lines are least-squares fits; the reward-shift panels also show the nominal $2H$ slope suggested by substituting the plotted proxy into the finite-horizon envelope, not a certified bound. Pearson r is 0.86 on cheetah-vel, 0.84 and 0.92 on reach and reach-wall in the raw units plotted here, 0.45 on ant-goal, and 0.16 and -0.16 on the hopper families. Fit definitions and statistics are in Table 5.

H.1 Dynamics-family metric

For the mass and friction families the total-variation transition term of equation 1 is vacuous, and Lemma 1-W provides a conditional route through next-state displacement. Table 6 measures that displacement on a shared pool of training transitions and finds a strong empirical near-linear relation with log-scale distance. Table 5 therefore uses $|\Delta \log \text{scale}|$ as a plotting proxy, not as the theorem's certified metric. Task scales form a forty-point log grid from 0.44 to 2.25; the training indices 10 to 29 span 0.67 to 1.48, and the held-out indices 5 to 9 and 30 to 34 have a log-scale distance of 0.04 to 0.21 to the nearest training task.

H.2 Expert cross-task regret

Proposition A.1 adds the expert cross-task regret κ_i to the extrapolation bound when the expert return E stands in for J^*. Table 7 reports it on the four MuJoCo families.

When κ_i is nonpositive the bound of Proposition A.1 applies with the measured quantities alone, and Table 7 shows this is family-dependent. On cheetah-vel every κ_i is nonpositive, so the bound holds as stated. On hopper-mass the light-side OOD experts beat the nearest training expert by 31 to 86 return, which means that E_i underestimates $J^*(\theta_i)$ by at least that amount, while the heavy side is nonpositive; hopper-friction shows the same asymmetry with large negative values on the high-friction side. Ant-goal is nonpositive on 8 of 10 tasks. Meta-World is omitted because its success

Table 6: Dynamics-family diagnostic for Lemma 1-W: next-state L_2 displacement $\|f_\theta(s,a) - f_{\theta'}(s,a)\|$ on a shared pool of 4000 training transitions, for all pairs of the 40 task scales. Mean displacement is strongly correlated with $|\Delta \log \text{scale}|$: Pearson correlation over all 780 pairs is 0.999 for mass and 0.975 for friction, and for each OOD task against the twenty training tasks its rounded minima are 1.000 and 0.999, respectively. Thus log-scale distance is an accurate empirical proxy on this sample, but it does not supply the global supremum or common value-Lipschitz coefficient required by the theorem. The last three columns give the measured quantities on held-out-to-nearest-training pairs: mean and maximum displacement and mean reward change.

env	pool	tasks	r (all pairs)	r (per task, min)	slope	mean $\|\Delta f\|_2$ (nearest)	max $\|\Delta f\|_2$ (nearest)	mean $\|\Delta r\|$ (nearest)
hopper-mass	4000	40	0.999	1.000	0.203	0.008 to 0.045	0.05 to 0.30	0.0005 to 0.0024
hopper-friction	4000	40	0.975	0.999	0.326	0.011 to 0.049	0.16 to 0.65	0.0005 to 0.0024

analysis uses the common ceiling 1 as a reference rather than treating a stored actor as an expert; by Corollary 1, no non-trivial task-metric bound is claimed for that success decomposition.

Table 7: Expert cross-task regret $\kappa_i = J_{\theta_i}(\pi^E_{\theta^*}) - J_{\theta_i}(\pi^E_{\theta_i})$, the OOD task's expert rolled on the nearest training task minus that task's own expert, $K = 5$ rollouts each. The lower side lists the five OOD tasks below the training range (indices 5 to 9) and the upper side the five above (30 to 34); for ant-goal, whose target goals all lie at radius 1.6, the two groups are the two angular halves, indices 20 to 24 and 25 to 29.

env	K	κ lower side	κ upper side	mean	median	$\kappa \le 0$
cheetah-vel	5	-95 to -19	-92 to -19	-55	-54	10/10
ant-goal	5	-316 to $+81$	-87 to $+31$	-54	-49	8/10
hopper-mass	5	$+31$ to $+86$	-29 to -6	$+19$	$+13$	5/10
hopper-friction	5	-30 to $+44$	-372 to -15	-108	-23	6/10

I SELECTION ACCURACY AGAINST CONTEXT SIZE

Theorem 2-A controls selection of the population likelihood minimiser i_{L} under i.i.d. sampling; it does not assert that i_{L} is the true task index. Figure 5 instead reports in-distribution task-recovery accuracy, where the true index is known, and is therefore a diagnostic rather than a direct test of the theorem. Accuracy rises with m and no curve of the thirty ends below its value at $m = 25$, though fourteen dip by a few points at one context size once they have saturated. On MuJoCo, hopper-mass CSRO goes from 0.44 at $m = 25$ to 0.94 at $m = 400$, hopper-friction CSRO from 0.46 to 0.96, ant-goal CSRO from 0.71 to 0.91, and cheetah-vel is at 0.99 or above from $m = 100$ on. On Meta-World, CSRO reaches 0.94 on reach, 0.90 on reach-wall and 0.89 on pick-place at $m = 200$, but only 0.68 on push and 0.55 on plate-slide, the contact families where reward likelihood is least discriminative. The softmax mass of the selected code, $w_{\max}$, exceeds 0.9 at $m = 200$ in every MuJoCo cell and in 14 of the 18 Meta-World cells, with the exceptions listed in Figure 5. It measures score concentration only: without knowing that the selected code is i_{L}, Lemma B does not turn it into a correctness certificate at deployment.

J ENVIRONMENTS, TASKS AND DATA COLLECTION

What a task is, and how the held-out split is made. Every family contains forty tasks and is split once into twenty training tasks, ten held-out tasks just outside the training construction, and ten farther-out tasks. This paper uses the training tasks and the first held-out band; Appendix E gives the family exclusion rule. Each held-out construction extends one designated task dimension while holding the remaining-coordinate construction fixed: speed for cheetah, radius for ant-goal, a physical scale for the dynamics families, and one dominant goal axis for Meta-World. The split is a property of the environment class, not of the evaluation: the framework's loader fills one replay buffer per split, so no held-out transition enters a training gradient. Table 8 gives the parameters, ranges and task-parameter distances or proxies.

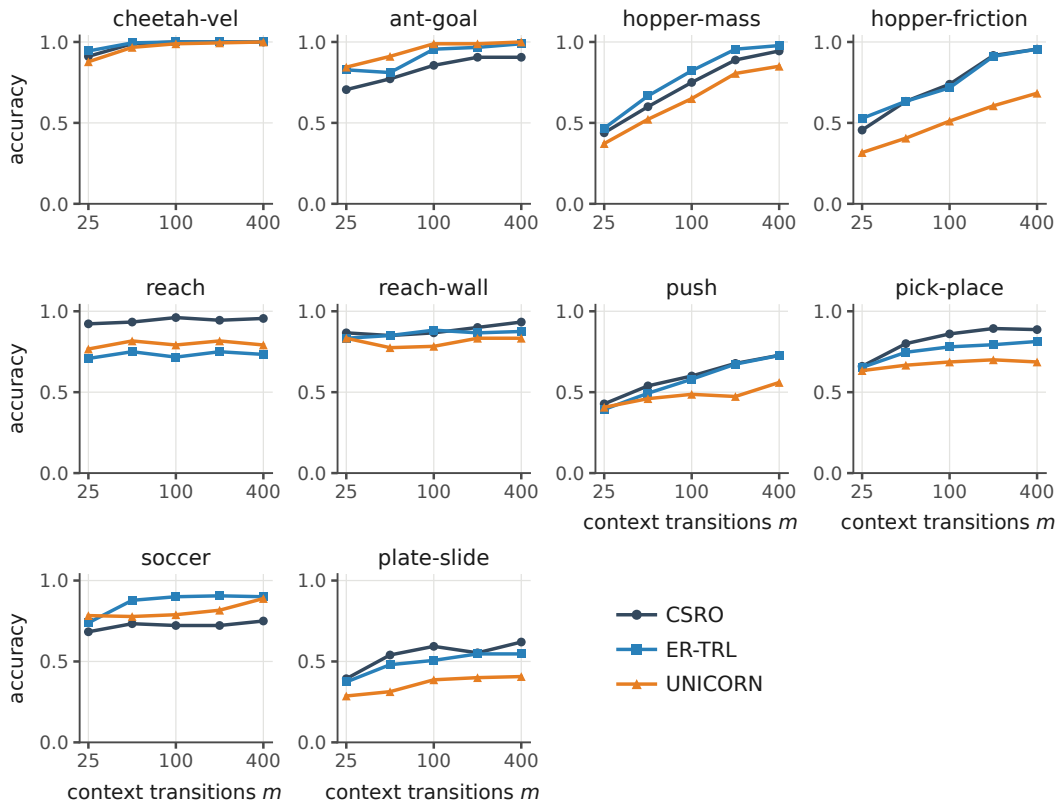


Figure 5: Selection accuracy against context size m on the training tasks: the probability that likelihood selection identifies the true task among twenty training codes from m offline transitions, averaged over tasks and seeds, one curve per algorithm. MuJoCo uses the window decoder with the full-transition score and Meta-World the all-tier decoder with the reward-only score, the pairing used in this diagnostic.

Table 8: Task parameters, splits and dimensions. *Distance to the hull* is a task-parameter distance or proxy associated with Section 3: absolute speed difference, radial goal displacement, $|\Delta \ln c|$ for dynamics families, or Euclidean Meta-World goal displacement (also in success-radius units). Only the cheetah quantity is an analytic reward-distance bound; the dynamics values are empirical proxies. d_z lists code dimensions for CSRO, ER-TRL and UNICORN.

family	task parameter	training tasks	held-out tasks	to the hull	obs/act	d_z
MuJoCo: dense reward, undiscounted return, $H = 200$, forty-point grid, training indices as given						
cheetah-vel	target speed v^*	1.1 to 3.0 (idx 10–29)	0.6–1.0, 3.1–3.5	0.1–0.5	20/6	20, 20, 5
ant-goal	goal (x, y), radius r	$r \in \{0.8, 1.2\}$, ten angles	$r = 1.6$, same angles	0.40	29/8	20, 20, 20
hopper-mass	mass scale $c = 1.5^e$	$c \in [0.674, 1.484]$ (10–29)	0.55–0.65, 1.55–1.83	0.04–0.21	11/3	20, 20, 40
hopper-friction	friction scale c	the same grid	the same grid	0.04–0.21	11/3	20, 20, 40
walker-mass	mass scale c	the same grid	the same grid	0.04–0.21	17/6	40, 40, 32
walker-friction	friction scale c	the same grid	the same grid	0.04–0.21	17/6	40, 40, 32
Meta-World: dense shaped reward plus binary success, $H = 200$, goal only, object start frozen per family, obs/act $36/4$, $d_z = 20$						
reach	goal height z, r=0.05	0.050 to 0.125	0.135 to 0.185	0.013–0.070	0.25–1.4 r	
reach-wall	goal height z, r=0.05	0.050 to 0.125	0.135 to 0.185	0.011–0.070	0.23–1.4 r	
pick-place	goal height z, r=0.07	0.050 to 0.090	0.100 to 0.150	0.013–0.065	0.18–0.93 r	
push	goal lateral x, r=0.05	-0.100 to -0.050	-0.040 to $+0.010$	0.010–0.061	0.21–1.2 r	
soccer	goal lateral x, r=0.07	-0.100 to -0.055	$+0.028$ to $+0.055$	0.083–0.111	1.19–1.6 r	
plate-slide	goal lateral x, r=0.07	-0.100 to -0.055	-0.045 to $+0.025$	0.010–0.081	0.15–1.2 r	

MuJoCo families. Cheetah-vel places forty target speeds on a linear grid from 0.1 to 4.0 and rewards $-|v - v^*|$ minus a control cost, so only the reward changes across tasks and $|v_a^* - v_b^*|$ is an analytic upper bound on the reward distance, attained at one-sided speeds. Ant-goal places twenty training goals on two half-circles of radius 0.8 and 1.2 at ten angles spanning $[-90°, 90°]$, and the held-out band is the same ten angles at radius 1.6: it is a pure radial extrapolation, and it is the one

family whose training indices are 0 to 19 rather than 10 to 29. The four dynamics families scale a physical property of the model by $c = 1.5^e$ with e on a forty-point grid from -2 to 2; hopper-mass and walker-mass scale link mass and inertia by the same factor, and the friction families scale the floor geom as well. The scale is not in the observation, so it can only be inferred from the transitions, and the unscaled model $c = 1$ is not itself a task. For these families, $|\Delta \ln c|$ is an empirical proxy for sampled next-state displacement, not the conditional Wasserstein metric itself (Table 6).

Meta-World families. A task is one frozen Meta-World goal vector. The object start is held at one nominal position per family, so the goal alone is the task parameter, and it is stripped from the observation the agent sees, which is therefore identical across tasks at the first step. The forty goals of a family are drawn once, under a fixed seed, by farthest-point sampling inside three axis-aligned boxes that differ only along one dominant axis: goal height for reach, reach-wall and pick-place, and goal lateral position for push, soccer and plate-slide. The other two axes have the same range in all three boxes, so the held-out shift is one-dimensional by construction. Success is the environment's own criterion, $\|x - g\| \le r$ at any step of the episode with x the gripper for the reach families and the manipulated object otherwise; episodes always run the full 200 steps and success is not absorbing. Five of the six bands are *adjacent*: the dead zone between the training and held-out boxes is about one centimetre, so the nearest held-out goals sit only 0.15 to 0.25 success radii outside the training hull, while the farthest sit at about $1.4\,r$. Soccer is the exception, with a dead zone of 8.3 centimetres and every held-out goal at $1.19\,r$ or beyond. In push and soccer the geometry is stronger than a shift: the object starts at $x = 0$, every training goal lies to its left, and the held-out goals straddle or lie to the right of it.

Why the main band stops where it does. This paper asks which code to deploy, which presupposes that some training code solves the task: on the main held-out band the best of twenty training codes reaches 0.88 to 1.00 (Table 15). In the farther band, measured only in the earlier metric diagnostic, this ceiling collapses to 0.00 on reach-wall and pick-place, 0.03 to 0.08 on reach, plate-slide and push, and 0.46 on soccer. We therefore exclude that band from the main diagnosis and method comparison. On the main band the encoder reaches 0.53 with oracle offline context and the framework channel 0.26 (Section 4), isolating a context failure despite the nearby code-set ceiling.

Data collection. For every task we train one SAC agent (Haarnoja et al., 2018) independently and record, at snapshots along its training curve, fifty stochastic and twenty deterministic 200-step evaluation episodes. These trajectories, rather than the SAC replay buffer or one final policy, provide varied behaviour quality. The training budget is 10^6 steps for cheetah-vel, hopper and walker, 3×10^6 for ant-goal, and from 5×10^5 to 2×10^6 by Meta-World family. The curated-window files retain stochastic rollouts from a late-snapshot window; push, pick-place and soccer also require deterministic success of at least 0.8. The loader streams these files into a circular per-task buffer capped at 2×10^5 transitions, so raw file counts above the cap are not simultaneously resident. The frozen checkpoints and window decoder use this curated source. The all-tier source instead contains every stored snapshot from 5×10^4 steps onward, including zero-success early snapshots; 18 to 88 snapshots are available per task, and each decoder run samples at most 80 files. Table 9 reports raw stored-file sizes and available snapshot counts. Held-out datasets are collected likewise for oracle-context evaluation and never used for training.

K Implementation details

Checkpoints and data. We use frozen checkpoints of CSRO, ER-TRL and UNICORN trained with each method's public code on the offline datasets of twenty training tasks per environment: on cheetah-vel and the hopper families the task indices 10 to 29 of a forty-point grid (velocities, log-spaced mass and friction multipliers), on ant-goal twenty goal positions on half-circles of radius 0.8 and 1.2 (indices 0 to 19), on Meta-World twenty goals inside the training box of each family. The datasets are collected by SAC experts (Haarnoja et al., 2018) and loaded by the framework's own loader, which keeps the stochastic evaluation rollouts of a curated window of SAC snapshots; on push, pick-place and soccer the loader additionally drops any snapshot whose deterministic success rate is below 0.8, and on the other three Meta-World families and on MuJoCo the window itself is the whole curation. Episodes have $H = 200$ steps and returns are undiscounted. Seeds are six

Table 9: Stored trajectory files selected by the curated-window rules. Episodes are 200 steps and counts are raw file counts before the loader's 2×10^5-transition circular-buffer cap. Ranges reflect task-dependent filtering. The fourth column totals twenty training tasks; the fifth is disk size for all forty. The last column gives available all-tier/window snapshots per task; a decoder run loads at most 80 all-tier files.

family	episodes/task	transitions/task	20 train tasks	bytes (40 tasks)	available all-tier/window
cheetah-vel	900	180 k	3.60 M	1.78 GB	19 / 18
ant-goal	2000	400 k	8.00 M	6.52 GB	40 / 40
hopper-mass	900	167–180 k	3.49 M	1.70 GB	19 / 18
hopper-friction	900	120–180 k	3.50 M	1.61 GB	19 / 18
walker-mass	900	155–175 k	3.34 M	2.01 GB	19 / 18
walker-friction	900	155–175 k	3.31 M	2.01 GB	19 / 18
reach	300	60 k	1.20 M	0.69 GB	37 / 6
reach-wall	250	50 k	1.00 M	0.58 GB	56 / 5
push	100–1200	20–240 k	4.38 M	2.53 GB	88 / 24
pick-place	200–1300	40–260 k	5.16 M	2.88 GB	88 / 26
soccer	450–900	90–180 k	3.50 M	1.67 GB	62 / 18
plate-slide	950	190 k	3.80 M	1.96 GB	42 / 19
MuJoCo, six families			25.24 M	15.63 GB	
Meta-World, six families			19.04 M	10.30 GB	

per cell on both benchmarks, 108 frozen checkpoints each. The environments, the splits and the collection are specified in Appendix J.

Training codes and geometry. Each training code $\bar{z}_i$ is the mean of five noise-free encoder inferences from 200-transition contexts drawn from D_i (for UNICORN the stochastic encoder is run at its mean); acting with $\bar{z}_i$ instead of a sampled-context code changes in-distribution success by -0.03 to $+0.02$ and cheetah-vel UNICORN return by -4.7, within seed noise. The code dimension is 20 on all six Meta-World families and on ant-goal, and 5 to 40 on the remaining MuJoCo families (Table 8). The encoder's OOD code $\hat{z}$ for the geometry terms is inferred from a matched 200-transition context of the OOD task's offline data, and the sensitivity probe evaluates the endpoint action displacement $\|\mu(s, \bar{z}_{\text{nearest}}) - \mu(s, \hat{z})\|_2$ on 256 states drawn from the deployed policy's own rollouts on θ^* (mean and maximum), which is the quantity of Lemma 2(b).

Channels of the decomposition. The offline channel infers the code from the OOD task's own offline data. The online channel is the framework's evaluation procedure: two exploration episodes with the prior code $z = 0$, then re-inference from the collected transitions. The non-prior channel is CSRO's exploration-decayed collection rule. The nearest channel acts with $\bar{z}_i$ of the parameter-nearest training task for three rollouts. The all-codes channel acts with every training code, for three rollouts each. B_{cv} is computed leave-one-out from these rollouts: the code is selected on two of the three and evaluated on the third. The expert value E is the return of the data-collecting SAC expert on the OOD task, or success 1 on Meta-World. The normalisation of the MuJoCo terms in Table 15 is $(x - F)/(E - F)$ with F the worst channel return observed in the environment.

Decoder. The decoder is a multilayer perceptron with two hidden layers of 200 units and SiLU activations that maps the normalised state, the action and the code to the mean of $(\Delta s, r)$, with a single learned log-variance vector shared across inputs (homoscedastic). It is trained for 6000 Adam steps at learning rate 10^{-3} with cosine annealing on batches of four training tasks times 128 transitions, with the code set to $\bar{z}_i$ or, with probability 0.3, to one of the five noise-free samples behind it. The all-tier decoder draws from up to 80 SAC snapshot files per training task, including low-return early checkpoints excluded by the curated window; the window decoder draws from the buffer source used for policy training. The policy and encoder are never retrained. Scores are sums of per-transition Gaussian log-likelihoods over the context: the reward coordinate alone for Meta-World, using its dense shaped reward, and all coordinates for every MuJoCo family. The binary Meta-World success signal is used separately by the keep and verification rules; scoring the full transition is the ablation in Table 12.

Exploration and selection. The main Meta-World rule uses five distinct training codes chosen by farthest-point sampling in code space, starting at the code nearest the centroid and greedily adding the code farthest from the selected set. Each is run for one 200-step episode. If one or more episodes

succeed, the rule keeps the successful exploration code with the largest episode return; it does not re-infer a code on this branch. If all five fail, their 1000 transitions form the context for $\hat{z}_{\text{enc}}$ and $\bar{z}_i$. Agreement between $\bar{z}_i$ and the training code nearest $\hat{z}_{\text{enc}}$ triggers the unverified shortcut. Otherwise $\hat{z}_{\text{enc}}$ receives one evaluation episode, and $\bar{z}_i$ receives one unless its failed exploration episode is already available. The rule prefers a successful candidate; if neither succeeds it compares the measured episode returns, breaking an exact tie toward $\hat{z}_{\text{enc}}$. The $K = 3$ and $K = 5$ excite-then-select results elsewhere in the appendix are auxiliary selection-head analyses, not the full main rule. Each OOD task is evaluated with six repeats of the entire procedure.

The main MuJoCo rule has no binary success gate. Its five exploration episodes comprise two with the prior code $z = 0$ and one with each of three farthest-point training codes. Its candidate set is the encoder code inferred from all five episodes, the framework's online code inferred from the first two, the top three likelihood codes, and every explored code. Duplicate codes are collapsed, exploration returns are reused, and each remaining candidate receives one episode; the largest measured return is committed. This gives 7–10 adaptation episodes and a mean of 9.4, excluding the final reported evaluation rollout. The earlier prior-only and bracketing selector variants are reported separately in Appendix G. The offline-context selectors of Table 19 use $n = 200$ transitions drawn from the OOD task's offline data, three draws per task; the snap selector is $\arg\min_i \|\hat{z} - \bar{z}_i\|$ and the combined score is $S_i/n - \lambda(\|\hat{z} - \bar{z}_i\|/\tau)^2$ with $\lambda = 2$ and τ a fixed code-distance scale, computed post hoc on the stored selector outputs.

Baselines and budget. The framework channels are evaluated with the framework's own code paths and, for the budget check, re-evaluated with more exploration episodes by overriding the number of exploration trajectories. More budget does not help them: on Meta-World the online channel is 0.252, 0.242 and 0.244 at two, three and seven exploration episodes, and on MuJoCo ten exploration episodes are *worse* than two in all six families (cheetah-vel -137 to -269, ant-goal -361 to -409, hopper-mass 478 to 378, hopper-friction 403 to 353, walker-mass 314 to 256, walker-friction 392 to 325). The gap the diagnosis measures is not a budget gap; the context those episodes produce is the problem, which is what Section 5 acts on. Our checkpoint-compatible IDAQ reimplementation follows its return-threshold variant: it samples a training-task hypothesis, rolls out the corresponding stored code, accepts an episode when it contains a Meta-World success, re-infers from accepted episodes, and falls back to the prior code if none is accepted. We omit the original model-boosting ensemble. The main comparison uses $E = 5$ and $E = 7$, as labelled in Table 13; $E = 3$ appears only as a budget ablation. This implementation preserves the hypothesis-rollout and return-filter mechanism but should not be read as an exact reproduction of every component of IDAQ. The budget-matched excite-then-select variant with $K = 2$ uses the first two farthest-point codes and reaches a mean success of 0.42 over the 18 cells. Selection alone does not use the acceptance rule and reaches 0.46.

L Additional tables and figures

Training-side neighbours. Five lines of work attack the same failure from the training side rather than at deployment. MACAW (Mitchell et al., 2021) meta-trains an advantage-weighted learner offline and adapts by gradient steps rather than by a code, so there is no code to verify. VariBAD (Zintgraf et al., 2020) conditions the policy on a belief over the task and learns an exploration that resolves it; with a frozen offline policy that exploration is not learned but chosen, which is what our excitation condition governs. DREAM (Liu et al., 2021) makes the same point from the training side of online meta-RL, decoupling exploration from exploitation because end-to-end training couples them badly. SMAC (Pong et al., 2022) names the shift our Meta-World diagnosis measures, that the context an offline meta-trained agent collects for itself is distributed unlike the data its encoder saw, and repairs it with a phase of online self-supervised training. Closest to our likelihood head, BATI (Ma et al., 2025) replaces the encoder with per-training-task embeddings and infers by maximising a dynamics likelihood, so that inference does not depend on the behaviour that produced the context; it selects among those embeddings without executing any of them, which is the step we add.

Two neighbouring test-time rules. Liu & Zhang (2026) select, among the per-transition encodings of the observed context, the code that best reconstructs that context. The candidates are read out of a context that may already have left the training distribution, and are scored by the very decoder

whose support that context has left; their mean is exactly what our encoder head returns, since the frozen encoders of all three backbones aggregate a context by averaging its per-transition posteriors. We instead draw candidates from codes that are in-support by construction and normally settle the choice with an executed episode. Rupf et al. (2026) choose the latent optimistically, maintaining a confidence ellipsoid over a reward-weight vector and picking the latent that maximises an upper confidence bound. That criterion needs the two linear structures a behaviour foundation model is trained to have—reward linear in known state features and return linear in the successor features of the policy indexed by the latent. Our checkpoints provide neither, so the OpTI-BFM acquisition function is not defined for them and we do not instantiate it.

What does not work, in full. On plate-slide the keep step is already 0.86 and the full rule 0.87, so later arbitration adds only 0.01. On cheetah-vel UNICORN, VeriCode improves the framework channel from -131 to -104, but the prior-based candidate already dominates the learned heads. Outcome comparison cannot beat the best supplied candidate: without the prior code, the return-based rule loses 28 to 82 return to the framework channel on cheetah-vel (Appendix F), a pattern consistent with Theorem 2-B. Proposition 3 applies to fixed candidates because success is repeatable in our harness; it does not cover the full adaptive rule or agreement shortcut. On MuJoCo, one-episode return noise is 8 to 140 against code spreads of 107 to 176, so Proposition 3-R retains its $\exp(-v\Delta^2/(2\Delta_G^2))$ error term.

Two design choices of the selection head. Both were forced by failures we observed. The decoder is *homoscedastic*, a Gaussian with a fixed per-coordinate width, because a code-dependent width can let a far code win under covariate shift through an inflated variance (Remark 6); and it is trained on *every snapshot tier* of the training data, including the low-return checkpoints the curated window excludes, because exploration visits low-reward states and a decoder that has never seen them gives unreliable scores there (*decoder coverage*). For reward-shift families we score with the reward coordinate alone (Remark 5).

Table 10: VeriCode inside and outside the training range on the same frozen checkpoints (Meta-World success; MuJoCo undiscounted return; ten tasks and six trials per cell); the OOD rows use the held-out band of Section 4 and Appendix J. In distribution the rule is never significantly below the encoder in any of the eighteen Meta-World cells. The Meta-World in-distribution IDAQ figure comes from the random-code run of the same campaign.

setting	IDAQ	encoder	likelihood	**VeriCode**
Meta-World (success)				
in distribution	0.59	0.60	0.72	**0.96**
out of distribution	0.42	0.45	0.46	**0.83**
MuJoCo (return)				
in distribution	236	234	264	**331**
out of distribution	136	159	200	**267**

Removing the two assumptions. Table 11 takes away, one at a time, the two things the rule relies on. With the stochastic actor success is no longer a deterministic function of the code, so Proposition 3 holds only in its exponential form, and the measurement shows how far that is from the deterministic case: a verification episode agrees with the deployment outcome in 0.780 of 10,462 pairs against 1.000 with the deterministic actor, and a code kept from exploration succeeds again in only 0.67 of the trials. The rule falls from 0.83 to 0.58, and every competitor falls with it. The ceiling falls too, which is the fairer way to read the row: rolling out all twenty training codes with the stochastic actor, the best of them reaches 0.89 mean success compared to 0.96 with the deterministic actor, so the rule keeps 0.65 of what is attainable down from 0.87. What fails is not the arbitration but the premise that a measured success predicts the next one. Removing the success signal instead costs less: committing by one-episode return reaches 0.71, at a budget of 8.4 episodes versus 5.5, because a dense return is a noisier comparison signal than a success bit and more candidates survive to be compared.

The decoder choices, measured. Section 5 argues for a homoscedastic decoder trained on every snapshot tier, and two rows of Table 12 put numbers on both halves. In this ablation campaign, a

Table 11: Robustness on the same 108 frozen checkpoints. Row two uses the stochastic actor, row three removes the success signal and commits by one-episode return, and row four reduces the exploration budget from five to three. Here *keep*=0.74 in row two is the in-sample exploration-hit rate, whereas VeriCode's 0.58 is success on the later deployment rollout.

setting	encoder	likelihood	IDAQ	keep	**VeriCode**
as reported (deterministic actor, success signal, E=5)	0.45	0.46	0.45	0.73	**0.83**
stochastic actor (Prop. 3 no longer exact)	0.41	0.45	0.42	0.74	**0.58**
no success signal: commit by one-episode return	0.28	0.46	–	–	**0.71**
budget E=3	0.38	0.43	0.38	0.54	**0.71**

heteroscedastic decoder lets a far code win by predicting a wide variance and lowers the likelihood head from 0.44 to 0.32 and the rule by 0.033 ± 0.009. Scoring the full transition instead of reward alone helps the head, 0.44 to 0.50, yet leaves the rule 0.010 ± 0.007 lower. A better head is not necessarily a better rule because the keep step decides most trials first.

How many training codes are needed. Two rows thin the candidate set while holding the decoder and normaliser fixed on all twenty training tasks, so only the codes the rule may commit to change. Halving the set to ten evenly spaced codes costs nothing, $+0.014 \pm 0.018$, and the keep step rises from 0.73 to 0.78 because five farthest-point codes cover half the set. Reducing it to five costs 0.144 ± 0.020. The fraction of trials in which some candidate works falls from 0.963 to 0.926 to 0.690, and VeriCode attains 87, 91 and 100 percent of those measured ceilings. The method needs a candidate set that covers the task range, not specifically twenty tasks.

The two MuJoCo settings. The last block varies the return-comparison rule. Doubling the comparison episodes per candidate spends 14.0 episodes instead of 9.4 and returns 263 against 267; increasing v therefore does not improve mean return in this experiment, although Proposition 3-R itself only gives a worst-case bound. Replacing the three farthest-point-code episodes with prior-code episodes costs 31 return: five prior episodes crowd out the behavioural spread useful for selection in these runs.

Table 12: Ablations of VeriCode, each a separate campaign over the same 108 frozen checkpoints, ten OOD tasks and six adaptation trials. *Keep* is excitation alone; *episodes* counts adaptation episodes before deployment; the last column gives MuJoCo return where measured. The single-factor rows change one ingredient, *heads alone* removes the keep step, and the last block is MuJoCo only.

	Meta-World (success)					MuJoCo
rule	keep	likelihood	hybrid	**VeriCode**	episodes	**VeriCode**
VeriCode (farthest-point codes, E=5, all-tier decoder)	0.73	0.46	–	**0.83**	5.5	**267**
random exploration codes instead	0.69	0.44	0.52	**0.79**	5.6	**254**
budget E=7 (farthest-point codes)	0.83	0.46	–	**0.88**	7.3	–
budget E=3 (random codes)	0.54	0.43	–	**0.71**	3.9	–
no success gate on the context (random codes)	0.68	0.46	–	**0.79**	5.6	–
decoder trained on the curated window (random codes)	0.68	0.35	0.46	**0.75**	5.5	–
heteroscedastic decoder (random codes)	0.68	0.32	–	**0.76**	5.6	–
full transition score, not reward only (random codes)	0.68	0.50	–	**0.78**	5.6	–
ten candidate codes instead of twenty	0.78	0.45	–	**0.85**	5.3	–
five candidate codes instead of twenty	0.68	0.40	–	**0.69**	5.2	–
heads alone, no keep step	–	0.44	0.52	0.60	6.6	251
MuJoCo only; VeriCode-R is 267 at 9.4 episodes						
two verification episodes per candidate	–	–	–	–	14.0	**263**
five prior-code exploration episodes, not two	–	–	–	–	10.0	**236**

Predicted confidence, in full. The natural repair for two complementary heads is to switch between them when one is confident, and every trial logs the statistics such a rule would use: the likelihood margin of the selected code over the encoder's code under the same decoder, the margin to the runner-up, the posterior mass of the selected code, the agreement of the per-episode argmaxes, the encoder's rank among the training codes and the number of successful exploration episodes. None works (Figure 7): their correlation with the sign of the head difference is at most 0.17, the best

global threshold reaches 0.44 to 0.49, and a threshold tuned per cell on the test tasks themselves 0.49 to 0.52, against 0.62 for a per-trial oracle over the same two heads. Choosing the head per family by leave-one-seed-out cross-validation reaches 0.47. Outcome comparison does not predict which head is right; it checks.

The two heads compared on the same context. The two heads are recorded in every VeriCode trial on the same gated context, a controlled comparison of inference alone. Over 6,480 paired trials both round to 0.44: likelihood is significantly ahead in four cells and behind in three. The gate itself adds nothing (0.44 against 0.42 for IDAQ), a per-trial oracle over the heads reaches 0.62, and choosing per family by leave-one-seed-out cross-validation reaches only 0.47. The complementarity lives at the trial level.

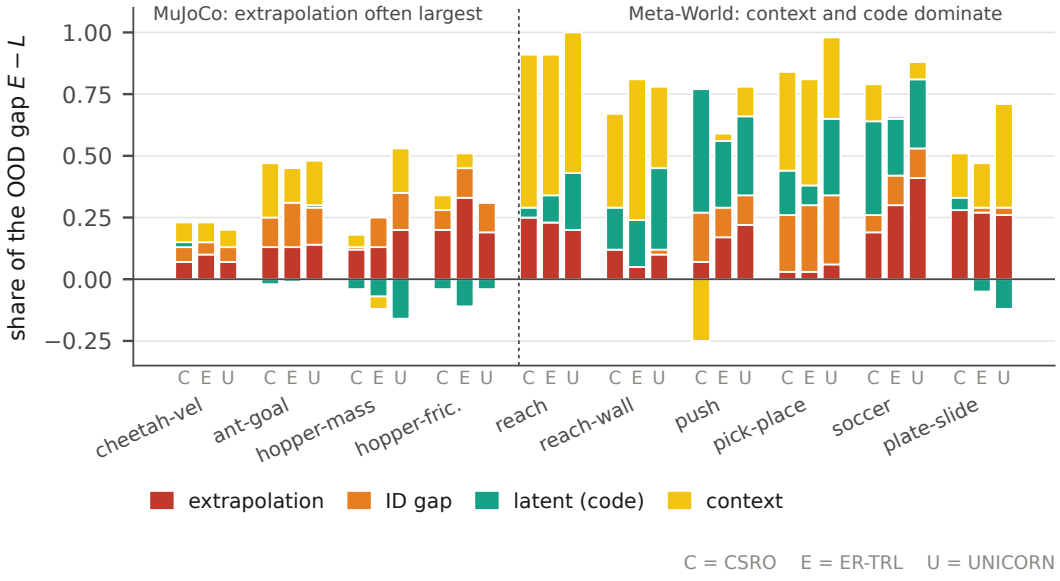


Figure 6: **Where the OOD gap goes.** The four terms of Theorem 1 on the ten-environment diagnosis grid as shares of $E - L$, one bar per algorithm (C = CSRO, E = ER-TRL, U = UNICORN); Table 15 gives the numbers. Extrapolation is largest in eight of twelve MuJoCo cells and the latent term is near zero; on Meta-World the context and latent terms dominate.

Table 13: Test-time adaptation on the six Meta-World families, OOD tasks: success rate over the ten OOD tasks per family, six adaptation trials and six seeds per cell, mean ± s.e.m. across seeds. *keep* is the excitation step of VeriCode alone (commit to an exploration code that succeeded), *oracle* the best of VeriCode's own candidates per trial, and the last column the paired per-seed difference between VeriCode and the best competitor of that cell (framework channel, IDAQ at either budget, either head, or the hybrid).

family	algorithm	$IDAQ_5$	$IDAQ_7$	encoder	likelihood	hybrid	keep	**VeriCode**	oracle	Δ
reach	csro	0.27 ±0.02	0.30 ±0.04	0.29 ±0.03	0.38 ±0.04	0.50 ±0.07	0.43 ±0.07	**0.70** ±0.09	0.70 ±0.09	+0.20 ± 0.05
reach	ertrl	0.27 ±0.02	0.32 ±0.04	0.36 ±0.02	0.35 ±0.03	0.38 ±0.07	0.67 ±0.05	**0.73** ±0.06	0.73 ±0.06	+0.35 ± 0.08
reach	unicorn	0.25 ±0.04	0.35 ±0.04	0.33 ±0.03	0.44 ±0.03	0.58 ±0.07	0.48 ±0.10	**0.73** ±0.10	0.73 ±0.10	+0.15 ± 0.07
reach-wall	csro	0.52 ±0.05	0.54 ±0.04	0.51 ±0.04	0.54 ±0.03	0.67 ±0.07	0.63 ±0.11	**0.87** ±0.06	0.87 ±0.06	+0.20 ± 0.04
reach-wall	ertrl	0.58 ±0.03	0.66 ±0.03	0.60 ±0.04	0.57 ±0.03	0.80 ±0.04	0.78 ±0.05	**0.93** ±0.03	0.93 ±0.03	+0.13 ± 0.03
reach-wall	unicorn	0.48 ±0.04	0.53 ±0.04	0.48 ±0.04	0.53 ±0.02	0.75 ±0.05	0.57 ±0.07	**0.83** ±0.03	0.83 ±0.03	+0.08 ± 0.04
push	csro	0.36 ±0.04	0.34 ±0.04	0.38 ±0.04	0.46 ±0.05	0.42 ±0.07	0.88 ±0.03	**0.92** ±0.03	0.92 ±0.03	+0.43 ± 0.12
push	ertrl	0.43 ±0.04	0.47 ±0.03	0.50 ±0.05	0.56 ±0.04	0.52 ±0.09	0.87 ±0.07	**0.93** ±0.05	0.95 ±0.03	+0.38 ± 0.06
push	unicorn	0.28 ±0.04	0.32 ±0.02	0.35 ±0.04	0.44 ±0.03	0.28 ±0.09	0.88 ±0.08	**0.92** ±0.05	0.93 ±0.03	+0.47 ± 0.05
pick-place	csro	0.39 ±0.05	0.46 ±0.06	0.42 ±0.07	0.33 ±0.03	0.55 ±0.13	0.70 ±0.08	**0.82** ±0.09	0.82 ±0.09	+0.27 ± 0.08
pick-place	ertrl	0.51 ±0.07	0.50 ±0.07	0.51 ±0.04	0.27 ±0.04	0.48 ±0.07	0.53 ±0.13	**0.67** ±0.10	0.67 ±0.10	+0.16 ± 0.07
pick-place	unicorn	0.31 ±0.04	0.35 ±0.03	0.33 ±0.04	0.28 ±0.07	0.37 ±0.10	0.73 ±0.10	**0.75** ±0.10	0.75 ±0.10	+0.38 ± 0.12
soccer	csro	0.34 ±0.06	0.34 ±0.05	0.31 ±0.04	0.35 ±0.08	0.32 ±0.07	0.95 ±0.03	**0.97** ±0.02	0.97 ±0.02	+0.61 ± 0.08
soccer	ertrl	0.34 ±0.09	0.30 ±0.06	0.32 ±0.07	0.36 ±0.04	0.28 ±0.09	0.68 ±0.10	**0.75** ±0.09	0.75 ±0.09	+0.39 ± 0.11
soccer	unicorn	0.17 ±0.02	0.21 ±0.02	0.24 ±0.02	0.46 ±0.05	0.25 ±0.07	0.82 ±0.10	**0.87** ±0.06	0.87 ±0.06	+0.41 ± 0.07
plate-slide	csro	0.63 ±0.04	0.64 ±0.04	0.60 ±0.03	0.54 ±0.07	0.67 ±0.04	0.82 ±0.04	**0.82** ±0.04	0.82 ±0.04	+0.15 ± 0.04
plate-slide	ertrl	0.73 ±0.06	0.74 ±0.04	0.67 ±0.04	0.54 ±0.03	0.73 ±0.04	0.85 ±0.04	**0.85** ±0.04	0.85 ±0.04	+0.11 ± 0.02
plate-slide	unicorn	0.71 ±0.02	0.73 ±0.03	0.68 ±0.01	0.60 ±0.03	0.83 ±0.05	0.90 ±0.04	**0.95** ±0.02	0.95 ±0.02	+0.12 ± 0.04

Table 14: Test-time adaptation on six MuJoCo families, OOD tasks: undiscounted return over the ten OOD tasks per family and six repeats, mean ± s.e.m. over six seeds. *Prior*, *enc* and *lik* are the three heads recorded in the same VeriCode trials (the re-inferred code from the two prior-code episodes, the re-inferred code from all five exploration episodes, and the top likelihood code). *Online* is the framework's channel; hybrid and IDAQ at $E = 5$ and $E = 10$ are separate campaigns on the same frozen checkpoints. VeriCode compares the prior, encoder and top-three likelihood codes by one-episode return and treats the exploration codes as candidates whose return is already measured.

env	algo	online	prior	enc	lik	hybrid	IDAQ E=5	IDAQ E=10	**VeriCode**
cheetah-vel	CSRO	-136±9	-140±10	-143±7	-136±10	-196±19	-271±1	-270±1	**-93±4**
	ER-TRL	-136±3	-139±3	-138±5	-146±8	-212±17	-267±1	-266±2	**-95±4**
	UNICORN	-131±4	-132±5	-166±10	-157±12	-243±4	-267±1	-267±1	**-104±8**
ant-goal	CSRO	-370±20	-363±5	-334±8	-243±8	-253±4	-459±31	-448±30	**-235±8**
	ER-TRL	-360±13	-360±9	-334±11	-305±20	-302±20	-378±7	-377±7	**-280±20**
	UNICORN	-380±12	-395±7	-363±11	-280±26	-279±28	-401±4	-400±6	**-259±27**
hopper-mass	CSRO	468±31	473±30	482±44	521±38	488±46	494±27	509±27	**552±33**
	ER-TRL	429±51	427±44	432±36	468±58	417±34	430±42	438±42	**497±45**
	UNICORN	329±58	299±46	447±47	451±51	493±37	472±22	494±31	**560±18**
hopper-friction	CSRO	414±11	395±17	400±22	433±14	405±12	403±10	406±12	**458±7**
	ER-TRL	342±37	347±35	351±43	351±36	365±24	354±25	357±29	**411±31**
	UNICORN	422±14	418±13	421±19	360±33	424±13	403±11	417±10	**469±8**
walker-mass	CSRO	260±22	256±8	259±18	334±35	291±14	280±5	262±8	**438±33**
	ER-TRL	325±44	316±42	306±26	353±19	302±15	302±31	285±12	**437±42**
	UNICORN	222±33	220±24	275±13	386±23	292±13	304±16	309±15	**465±11**
walker-friction	CSRO	345±39	338±24	330±18	373±27	362±27	345±25	324±18	**529±28**
	ER-TRL	397±37	408±39	361±60	419±29	375±55	396±43	410±43	**512±31**
	UNICORN	274±33	296±35	285±20	413±29	326±19	308±16	280±20	**542±24**
mean (18 cells)		151	148	159	200	170	136	137	**267**

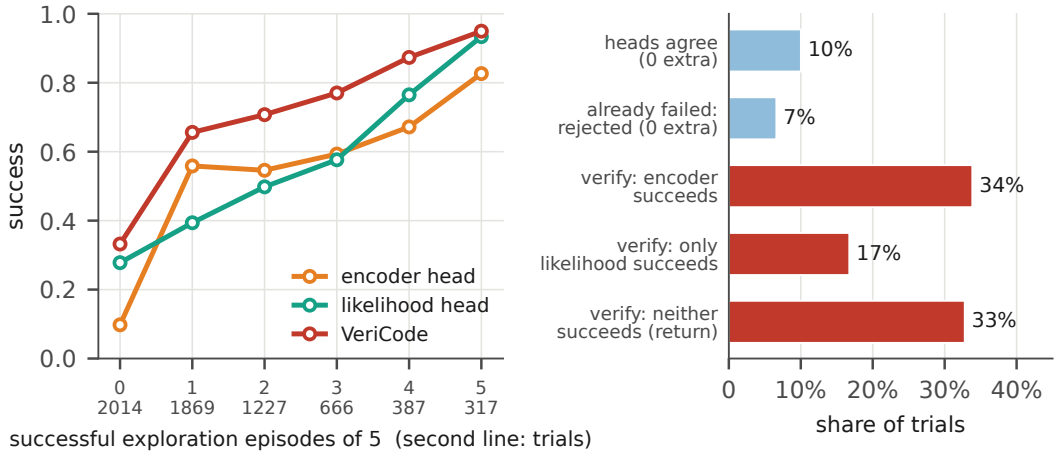


Figure 7: **Mechanism and budget (Meta-World).** Left: success of the two heads and of VeriCode against the number of successful exploration episodes. Right: how the trials resolve under the rule that arbitrates the heads without the keep step; 17 percent commit without verification and 83 percent spend two verification episodes, a mean budget of 6.6, against 5.5 for VeriCode, which commits on a measured exploration success in 73 percent of the trials.

Two ingredients of the selection head. Table 16 in the appendix crosses, for the selection head, the decoder's training data (curated window or all snapshot tiers) with the exploration (prior code or three farthest-point codes). The four cells are 0.28, 0.33, 0.38 and 0.46, so exploration alone contributes $+0.05$ to $+0.08$, decoder coverage alone $+0.10$ to $+0.13$, and the interaction is $+0.03$. The two changes target different quantities in the analysis: exploration targets the excitation gap of Theorem 2-B, while broader decoder coverage targets the errors represented by δ and κ in Theorem 2-A; these global quantities are not measured here (Table 17, Figure 8). The encoder re-inferred from all three excitation episodes stays at 0.25, so an excited context does not help the amortised encoder by itself; the benefit of restricting context to accepted successful episodes is consistent with IDAQ's design.

Reading the online ablation. Figure 8 places the five practical rules and three reference levels on one axis. The framework's own online channel reaches 0.26 and its non-prior variant 0.23. Prior-code exploration with the curated-window decoder reaches 0.28, and diverse excitation with

Table 15: Where the OOD gap goes: the four terms of Theorem 1 on the ten-environment grid (OOD split, averaged over OOD tasks and seeds), with the channels E, L, N, O of Section 3 and E equal to success 1 on Meta-World. In the four MuJoCo families the terms are returns normalised per environment as $(x - F)/(E - F)$ with F the worst channel return observed, in the six Meta-World families they are success-rate differences. The dominant term carries its share of $E - L$ in parentheses. The ceilings are in raw units, with B_{cv} the cross-validated best of the twenty training codes and $B_{\mathrm{cv}}{>}O$ the fraction of OOD tasks on which it beats O.

		normalised decomposition of $E - L$						ceilings (raw units)				
env	algo	$E-L$	Δ_{extrap}	ε	Δ_{latent}	Δ_{ctx}	dominant (share)	E	O	N	B_{cv}	$B_{\mathrm{cv}}{>}O$
cheetah-vel	CSRO	0.23	0.07	0.06	0.02	**0.08**	ctx (0.36)	−26	−99	−91	−93	0.52
	ER-TRL	0.23	**0.10**	0.05	0.00	0.08	extrap (0.43)	−26	−100	−99	−93	0.60
	UNICORN	0.21	**0.07**	0.06	0.00	0.07	extrap (0.34)	−26	−95	−93	−95	0.45
ant-goal	CSRO	0.46	0.13	0.12	−0.02	**0.22**	ctx (0.49)	−78	−223	−235	−241	0.42
	ER-TRL	0.43	0.13	**0.18**	−0.01	0.14	ε (0.42)	−78	−262	−271	−259	0.60
	UNICORN	0.48	0.14	0.15	0.01	**0.18**	ctx (0.37)	−78	−266	−257	−249	0.50
hopper-mass	CSRO	0.14	**0.12**	0.01	−0.04	0.05	extrap (0.82)	613	553	527	580	0.63
	ER-TRL	0.13	**0.13**	0.12	−0.07	−0.05	extrap (1.00)	613	502	455	549	0.67
	UNICORN	0.38	**0.20**	0.15	−0.16	0.18	extrap (0.54)	613	497	400	575	0.73
hopper-friction	CSRO	0.30	**0.20**	0.08	−0.04	0.06	extrap (0.67)	587	449	427	472	0.70
	ER-TRL	0.39	**0.33**	0.12	−0.11	0.06	extrap (0.84)	587	396	328	435	0.75
	UNICORN	0.27	**0.19**	0.12	−0.04	0.00	extrap (0.70)	587	438	411	480	0.77
reach	CSRO	0.92	0.25	0.00	0.04	**0.62**	ctx (0.68)	1.00	0.71	0.75	0.95	0.30
	ER-TRL	0.92	0.23	0.00	0.11	**0.57**	ctx (0.63)	1.00	0.66	0.77	0.92	0.28
	UNICORN	1.00	0.20	0.00	0.23	**0.57**	ctx (0.57)	1.00	0.57	0.80	0.90	0.37
reach-wall	CSRO	0.67	0.12	0.00	0.17	**0.38**	ctx (0.58)	1.00	0.72	0.88	0.95	0.27
	ER-TRL	0.82	0.05	0.00	0.19	**0.57**	ctx (0.70)	1.00	0.76	0.95	1.00	0.25
	UNICORN	0.77	0.10	0.02	**0.33**	0.33	latent (0.42)	1.00	0.56	0.88	0.98	0.48
push	CSRO	0.52	0.07	0.20	**0.50**	−0.25	latent (0.97)	1.00	0.23	0.73	1.00	0.80
	ER-TRL	0.60	0.17	0.12	**0.27**	0.03	latent (0.44)	1.00	0.43	0.70	1.00	0.70
	UNICORN	0.78	0.22	0.12	**0.32**	0.12	latent (0.41)	1.00	0.35	0.67	0.97	0.68
pick-place	CSRO	0.85	0.03	0.23	0.18	**0.40**	ctx (0.47)	1.00	0.55	0.73	1.00	0.50
	ER-TRL	0.82	0.03	0.27	0.08	**0.43**	ctx (0.53)	1.00	0.62	0.70	0.98	0.40
	UNICORN	0.97	0.06	0.28	0.31	**0.33**	ctx (0.34)	1.00	0.36	0.67	0.88	0.55
soccer	CSRO	0.80	0.19	0.07	**0.38**	0.15	latent (0.48)	1.00	0.35	0.73	1.00	0.73
	ER-TRL	0.65	**0.30**	0.12	0.23	0.01	extrap (0.46)	1.00	0.36	0.58	0.98	0.67
	UNICORN	0.88	**0.41**	0.12	0.28	0.07	extrap (0.46)	1.00	0.19	0.47	0.98	0.87
plate-slide	CSRO	0.52	**0.28**	0.00	0.05	0.18	extrap (0.55)	1.00	0.67	0.72	0.97	0.33
	ER-TRL	0.42	**0.27**	0.02	−0.05	0.18	extrap (0.64)	1.00	0.77	0.72	0.92	0.18
	UNICORN	0.58	0.26	0.03	−0.12	**0.42**	ctx (0.73)	1.00	0.84	0.72	0.95	0.15

encoder re-inference 0.26. Diverse excitation with the all-tier decoder, our selection head, reaches 0.46. Against reference levels of 0.73 for the parameter-nearest code, 0.53 for oracle offline context and 0.96 for B_{cv}, the largest practical gain appears when excitation (Theorem 2-B) and decoder coverage (Theorem 2-A) are combined.

The two factors are nearly additive. Table 16 crosses the same two ingredients in a 2×2 design. The four cells are 0.28, 0.33, 0.38 and 0.46, so the main effects are $+0.07$ for exploration and $+0.12$ for coverage with an interaction of $+0.03$: close to additive. This pattern is consistent with the two components acting on different quantities in the analysis—the excitation gap $\Delta_{\min}$ and the slacks δ and κ—but it is not a measurement of those quantities. The encoder applied to the same contexts stays at 0.24 and 0.25, and the non-prior channel with its two exploration episodes is 0.23 on the same checkpoints.

Table 16: Two-factor decomposition of the Meta-World online gain, averaged over 18 environment $\times$ algorithm cells. Rows vary decoder data: the curated window (with a success ≥ 0.8 filter only for push, pick-place and soccer) or all stored snapshot tiers, with policy and encoder fixed. Columns vary excitation: two prior-code episodes or three farthest-point-code episodes. Bold marks our selection head.

decoder \ excitation	prior code $z{=}0$	diverse codes ($K{=}3$)	exploration effect
curated-window decoder	0.28	0.33	+0.05
all-tier decoder	0.38	**0.46**	+0.08
coverage effect	+0.10	+0.13	+0.03
encoder on the same context	0.24	0.25	+0.01

The two factors per family. Table 17 shows that the factors are not uniform. Reach-wall rises from 0.14 to 0.42 when its two prior-code episodes are replaced by diverse excitation with the window decoder fixed. Reach and pick-place are best when both factors are present; push gains most

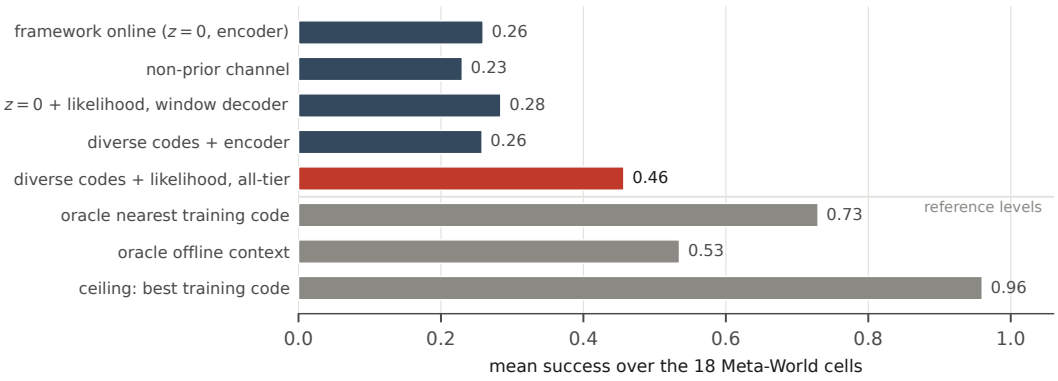


Figure 8: Online ablation on Meta-World: mean success over the 18 environment × algorithm cells of the OOD split, on the same frozen checkpoints throughout. The five practical rules are above the horizontal divider and three reference levels below it.

from all-tier decoder coverage. Plate-slide's selector stays near 0.5 across conditions even though its encoder changes from 0.43 to 0.56, and soccer is best under the prior-plus-window selector. The aggregate encoder means, 0.24 and 0.25, therefore hide family-level movement rather than establish a consistent encoder gain.

Table 17: Per-family cells of the two-factor design of Table 16: success rate on the OOD split, algorithms pooled, with the encoder re-inferred from the same context in the last two columns. Bold marks the best of the four selector cells in each row.

	window decoder		all-tier decoder		encoder, same context	
family	z=0	diverse	z=0	diverse	z=0	diverse
reach	0.10	0.14	0.23	**0.36**	0.06	0.06
reach-wall	0.14	0.42	0.49	**0.64**	0.28	0.23
push	0.28	0.41	**0.56**	0.43	0.34	0.27
pick-place	0.24	0.18	0.21	**0.41**	0.09	0.14
soccer	**0.42**	0.34	0.28	0.41	0.22	0.24
plate-slide	0.52	0.49	**0.52**	0.52	0.43	0.56
mean	0.28	0.33	0.38	0.46	0.24	0.25

The channels of Table 18. The table places the frozen checkpoints in and out of distribution under seven diagnostic channels. *ID* is the training-task value with offline context. *OOD offline* gives the OOD task's own offline data as context, an oracle the deployed agent never has; *OOD online* is the framework's default test-time channel, two exploration episodes with the prior code $z = 0$ followed by re-inference. *select* is an earlier excite-then-select head, not the full VeriCode rule: on Meta-World it uses $K = 3$, and on MuJoCo it uses diverse-code exploration from the second selector wave. The latter is retained as a diagnostic even though it is harmful on cheetah-vel; the precursor prior-code and bracketing variants are in Table 4, and the main return-comparison rule is in Table 14. *snap* projects the offline-context code onto the nearest training code and *nearest* takes the parameter-nearest training code, so the two isolate geometry from likelihood. *best* is B_{cv}, the leave-one-out best training code; on MuJoCo the leave-one-out estimate carries no seed s.e.m. and is given as a mean, and the naive in-sample best, which is selection-biased upward, is not shown.

What the table shows is the asymmetry between the two benchmarks. On Meta-World the online channel collapses relative to oracle offline context, from 0.71 to 0.08 on reach for CSRO and from 0.55 to 0.15 on pick-place, so a large share of the OOD loss comes from the collected context. On MuJoCo the same comparison is generally smaller in relative terms, which is the measured form of the regime split of Section 4.

The two benchmarks read differently. The Meta-World half of Table 18 is the regime the method targets: performance drops when context is self-collected rather than supplied offline, and the deployable *online* and *select* columns sit below the oracle columns. On MuJoCo, the oracle-context

Table 18: In-distribution and out-of-distribution performance of the frozen checkpoints; mean ± seed s.e.m., returns for MuJoCo and success rate for Meta-World. ID is the training-task value with offline context, and the OOD columns are the held-out split. The channels are defined in the text above; bold marks the better of *online* and *select*, the only two a deployed agent can run, since the remaining columns all use information it does not have.

			ID	OOD					
env	algo	seeds	offline ctx	offline ctx	online (z=0)	select	snap	nearest	B_{cv}
MuJoCo (return)									
cheetah-vel	CSRO	6	−47±3	−99±4	**−140±10**	−163±22	−98±4	−91±4	−93
	ER-TRL	6	−50±5	−100±3	**−140±5**	−171±20	−103±6	−99±10	−93
	UNICORN	6	−51±2	−95±2	**−131±4**	−213±25	−97±3	−93±2	−95
ant-goal	CSRO	6	−116±6	−223±7	−361±6	**−333±17**	−227±4	−235±13	−241
	ER-TRL	6	−150±19	−262±25	**−346±12**	−371±19	−257±24	−271±24	−259
	UNICORN	6	−141±21	−266±24	−376±9	**−324±23**	−249±28	−257±25	−249
hopper-mass	CSRO	6	546±21	553±15	**521±42**	455±40	509±37	527±24	580
	ER-TRL	6	528±22	502±27	**529±27**	427±36	441±58	455±45	549
	UNICORN	6	532±23	497±29	384±39	**450±43**	517±22	400±52	575
hopper-friction	CSRO	6	560±10	449±16	**415±15**	411±15	435±17	427±20	472
	ER-TRL	6	520±25	396±22	**360±35**	325±31	330±47	328±46	435
	UNICORN	6	539±19	438±18	**433±8**	401±28	412±14	411±55	480
Meta-World (success rate)									
reach	CSRO	6	0.90±0.01	0.71±0.03	0.08±0.03	**0.32±0.07**	0.78±0.05	0.75±0.02	0.95
	ER-TRL	6	0.91±0.03	0.66±0.06	0.08±0.05	**0.35±0.08**	0.73±0.06	0.77±0.02	0.92
	UNICORN	6	0.90±0.01	0.57±0.04	0.00±0.00	**0.40±0.03**	0.60±0.04	0.80±0.04	0.90
reach-wall	CSRO	6	0.90±0.03	0.72±0.05	0.33±0.04	**0.60±0.05**	0.80±0.04	0.88±0.03	0.95
	ER-TRL	6	0.92±0.02	0.76±0.06	0.18±0.07	**0.67±0.07**	0.82±0.05	0.95±0.02	1.00
	UNICORN	6	0.89±0.05	0.56±0.05	0.23±0.09	**0.65±0.06**	0.58±0.05	0.88±0.05	0.98
push	CSRO	6	0.80±0.03	0.23±0.04	**0.48±0.14**	0.30±0.09	0.40±0.06	0.73±0.07	1.00
	ER-TRL	6	0.75±0.04	0.43±0.09	0.40±0.12	**0.52±0.06**	0.67±0.08	0.70±0.09	1.00
	UNICORN	6	0.85±0.03	0.35±0.05	0.22±0.14	**0.47±0.09**	0.48±0.07	0.67±0.06	0.97
pick-place	CSRO	6	0.82±0.04	0.55±0.04	0.15±0.07	**0.53±0.08**	0.68±0.10	0.73±0.08	1.00
	ER-TRL	6	0.78±0.07	0.62±0.09	0.18±0.08	**0.35±0.11**	0.57±0.04	0.70±0.04	0.98
	UNICORN	6	0.79±0.10	0.36±0.05	0.02±0.03	**0.35±0.11**	0.50±0.07	0.67±0.10	0.88
soccer	CSRO	6	0.87±0.04	0.35±0.09	0.20±0.07	**0.47±0.06**	0.50±0.06	0.73±0.04	1.00
	ER-TRL	6	0.82±0.04	0.36±0.07	**0.35±0.11**	0.32±0.03	0.45±0.11	0.58±0.05	0.98
	UNICORN	6	0.85±0.03	0.19±0.07	0.12±0.06	**0.45±0.08**	0.45±0.14	0.47±0.11	0.98
plate-slide	CSRO	6	0.98±0.02	0.67±0.04	0.48±0.09	**0.50±0.06**	0.65±0.04	0.72±0.04	0.97
	ER-TRL	6	0.98±0.01	0.77±0.06	**0.58±0.09**	0.57±0.04	0.77±0.05	0.72±0.02	0.92
	UNICORN	6	0.99±0.01	0.84±0.03	0.42±0.13	**0.48±0.06**	0.80±0.04	0.72±0.03	0.95

encoder exceeds the online channel by 36–41 return on cheetah-vel, 5–36 on hopper-friction and 84–138 on ant-goal; hopper-mass is mixed, from −27 to 113. The online channel beats the diverse-code selector in nine of twelve MuJoCo cells. This is consistent with the two-sided ambiguity proved by Proposition B.2 for its simplified symmetric-reward setting and motivates the prior-code candidate used in Table 4.

The *snap* column is worth reading against *nearest*. Snap projects the encoder's oracle-context code onto the training-code set and nearest takes the code of the parameter-nearest training task, so the two differ only in whether the projection is made in code space or in task space. The task-space choice is the better of the two in fifteen of the eighteen Meta-World cells, by up to 0.33 on push and 0.30 on reach-wall, and the three exceptions are the cells where the encoder is strongest. So the encoder's code, even with an oracle context, usually does not land nearest to the right training code in code space. That gap is an empirical manifestation of the latent term in Theorem 1, and it is why the method draws its candidates from the training codes instead of from wherever the encoder points.

Selectors on an oracle context. Table 19 removes the exploration problem entirely by giving every selector 200 transitions of the OOD task's own offline data, so what remains is the inference rule alone. *Encoder* is the framework's amortised inference O. *Likelihood (window)* scores the twenty training codes with a decoder trained on the curated window and the full transition. *Likelihood (all-tier)* uses the decoder trained on every snapshot tier and the reward-only score; this is the decoder used by our online method. The window column comes from the first selector wave, and the rest from the second wave, in which the full and reward-only scores coincide. *Snap* is $\arg\min_i \|\hat{z} - \bar{z}_i\|$,

the projection of the encoder's code onto the training-code set. *Oracle nearest* is the parameter-nearest training code and *ceiling* is B_{cv}.

Decoder coverage lifts likelihood selection from 0.50 to 0.56, but it remains below the geometry-only snap at 0.62. Every practical selector also remains below the parameter-nearest oracle at 0.73 and the ceiling at 0.96. Thus an oracle offline context does not eliminate the measured ranking gap for the scoring rules tested here, motivating direct verification.

Table 19: Offline-context selectors on Meta-World OOD tasks, success rate. Each selector receives $n = 200$ context transitions drawn from the OOD task's own offline data, so this axis isolates the inference rule from the exploration problem; n_{ctx} counts OOD task $\times$ seed $\times$ context-draw rows, three draws per task. The seven selectors are defined in the text above; bold marks the best of the four practical ones, the last two columns being oracles. The lower panel pools the three algorithms per family.

env	algo	n_{ctx}	encoder	lik. (window)	lik. (all-tier)	snap	oracle nearest	ceiling B_{cv}
reach	CSRO	180	0.71	0.43	0.55	**0.78**	0.75	0.95
	ER-TRL	180	0.66	0.49	**0.73**	0.73	0.77	0.92
	UNICORN	180	0.57	0.54	**0.63**	0.60	0.80	0.90
reach-wall	CSRO	180	0.72	0.69	**0.84**	0.80	0.88	0.95
	ER-TRL	180	0.76	0.72	0.81	**0.82**	0.95	1.00
	UNICORN	180	0.56	0.67	**0.74**	0.58	0.88	0.98
push	CSRO	180	0.23	0.42	**0.51**	0.40	0.73	1.00
	ER-TRL	180	0.43	0.57	0.56	**0.67**	0.70	1.00
	UNICORN	180	0.35	0.38	0.47	**0.48**	0.67	0.97
pick-place	CSRO	180	0.55	0.34	0.25	**0.68**	0.73	1.00
	ER-TRL	180	**0.62**	0.29	0.32	0.57	0.70	0.98
	UNICORN	180	0.36	0.27	0.31	**0.50**	0.67	0.88
soccer	CSRO	180	0.35	0.30	0.39	**0.50**	0.73	1.00
	ER-TRL	180	0.36	0.30	0.33	**0.45**	0.58	0.98
	UNICORN	180	0.19	0.37	0.39	**0.45**	0.47	0.98
plate-slide	CSRO	180	0.67	0.64	**0.69**	0.65	0.72	0.97
	ER-TRL	180	**0.77**	0.73	0.76	0.77	0.72	0.92
	UNICORN	180	0.84	0.77	**0.87**	0.80	0.72	0.95
mean (18 cells)			0.54	0.50	0.56	0.62	0.73	0.96

family	encoder	lik. (window)	lik. (all-tier)	snap	oracle nearest	ceiling B_{cv}
reach	0.65	0.49	0.64	**0.71**	0.77	0.92
reach-wall	0.68	0.69	**0.80**	0.73	0.91	0.98
push	0.34	0.46	0.51	**0.52**	0.70	0.99
pick-place	0.51	0.30	0.29	**0.58**	0.70	0.96
soccer	0.30	0.32	0.37	**0.47**	0.59	0.99
plate-slide	0.76	0.71	**0.78**	0.74	0.72	0.94
mean	0.54	0.50	0.56	0.62	0.73	0.96

What the oracle-context axis shows. Table 19 gives every rule 200 transitions from the OOD task's offline dataset, removing online exploration from this comparison. The best tested practical rule is the geometry-only projection at 0.62 and likelihood reaches 0.56; the unavailable parameter-nearest code reaches 0.73 and the measured code-set ceiling 0.96. The gap is therefore not eliminated by the scoring rules tested here, even with oracle offline context.

This is why the method spends episodes: verification replaces predicted ranking with a target-task measurement. With candidates restricted to the measured training-code set, no rule can exceed its observed 0.96 ceiling on these trials. VeriCode's 0.83 is 0.87 of that ceiling with self-collected context, against 0.65 for the best tested rule here with oracle offline context.