Response-Aware Exploration for Continuous Control Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

In reinforcement learning, exploration with sparse rewards and continuous action spaces on long-horizon tasks remains difficult, and a common strategy is to perturb the policy's actions during exploration. However, diverse action perturbations do not necessarily lead to diverse behavioral responses. Existing methods may encourage behavioral diversity indirectly, but few derive the perturbation directions themselves from the behavioral responses they actually produce. In this paper, we propose Discovered Action-Response Eigenbasis (DARE), a method that derives exploration directions by comparing perturbed and unperturbed rollouts from the same initial state and applies temporally correlated perturbations along those directions. DARE is reward-free, and is designed as an agile method that can be integrated into existing RL pipelines with minimal changes. Experimental results show that DARE outperforms representative action-space perturbation and temporally structured exploration baselines on sparse-reward tasks in Maniskill. DARE enables exploration to reach task-progress states more often, achieving up to 504% as many reaches as the strongest evaluated baseline, while remaining effective across different RL algorithms.

1 Introduction

Exploration remains a fundamental challenge in long-horizon continuous-control tasks with sparse rewards. Sparse feedback provides little guidance before successful outcomes are reached, particularly in tasks that require long sequences of coordinated actions (Amin et al., 2021; Eberhard et al., 2023). As a result, policies tend to collect experience repeatedly around their current behaviors without discovering more rewarding behaviors. Acquiring informative experience beyond the current policy's behavior distribution is therefore essential for effective exploration, especially when dense rewards, target-domain expert actions, motion planners, or other forms of external guidance are unavailable.

Existing research addresses this problem at three complementary levels, distinguished by where exploration intervenes: *objective*, *behavior*, and *controller*. At the *objective level*, exploration is encouraged by prioritizing potentially informative experiences, either through additional intrinsic incentives or by directing exploration toward selected regions of the state or goal space (Schäfer et al., 2022; Houthooft et al., 2016; Pitis et al., 2020; Pong et al., 2020). At the *behavior level*, higher-level behavioral abstractions such as skills, options, or latent behaviors are used, using which exploration is performed. They alter how agents interact with the environment over extended horizons (Yu et al., 2024; Eysenbach et al., 2019; Sharma et al., 2020). Finally, at the *controller level*, exploration intervenes directly in action generation, such as an exploration objective, a target state, or a behavioral abstraction to prescribe the deviation (Plappert et al., 2018; Zhang and Van Hoof, 2021; Raffin et al., 2022; Eberhard et al., 2023; Chiappa et al., 2023). Among these, controller-level exploration is particularly attractive because it can directly diversify behavior during interaction while remaining largely complementary to objective- and behavior-level methods.

However, applying different action perturbations does not necessarily lead to different behaviors, since distinct perturbation directions can still produce similar or highly uneven behavioral responses (Baram et al., 2021; Chiappa et al., 2023). This shifts the problem

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.

from simply diversifying perturbations in action space to finding directions that actually lead to distinct behavioral outcomes.

To address this issue, we introduce *Discovered Action-Response Eigenbasis* (DARE), which chooses action perturbations based on the behavioral changes they actually produce. DARE first discovers action directions that lead to different behavioral responses, and then combines these directions while assigning greater exploration weight to those with larger effects. The resulting bias is added directly to the policy action. DARE requires no target-domain expert actions, motion planners, dense rewards, or externally prescribed action directions.

We evaluate DARE on sparse-reward continuous-control tasks in ManiSkill, where reward is only obtained after long sequences of coordinated actions, providing little guidance during exploration. DARE remains competitive across tasks, with its clearest gains appearing on milestones that require sustained progress in a consistent behavioral direction. By perturbing only executed actions, DARE can be used across learning algorithms while remaining compatible with higher-level exploration methods.

2 Related Works

Existing controller-level methods for exploration in RL can be grouped by *how long* a perturbation is maintained: step-wise, segment-wise, or trajectory-wise. (Amin et al., 2021; Eberhard et al., 2023; Zhang and Van Hoof, 2021).

Trajectory-wise exploration. Trajectory-wise exploration aims to keep exploratory behavior coherent over long horizons. A representative approach is parameter-space noise, which perturbs policy parameters for an episode so that the resulting behavioral change persists throughout the rollout (Plappert et al., 2018). Episodic RL takes a related view by operating on parameters of complete movement trajectories, with recent variants improving data efficiency through step-level updates or off-policy value learning over action sequences (Otto et al., 2023; Li et al., 2024; 2025). Other methods lie between the step-wise and episode-wise extremes. Generalized Exploration allows policy parameters to evolve over time (van Hoof et al., 2017), and Deep Coherent Exploration brings this idea to deep policies through latent policy parameters (Zhang and Van Hoof, 2021). Overall, these methods mainly shape how long exploratory deviations remain coherent. DARE focuses on a different question: which action-space directions should be explored, based on the behavioral responses that those perturbations actually produce.

Segment-wise exploration. Segment-wise methods structure exploration over a finite temporal window. Locally Persistent Exploration introduces short-term directional memory through trajectory-dependent actions (Amin et al., 2021). More closely related to our setting, gSDE holds a state-dependent noise matrix for several steps before resampling, producing temporally coherent perturbations (Raffin et al., 2022). Lattice further introduces latent and action-space noise matrices, combining temporal persistence with cross-dimensional correlations (Chiappa et al., 2023). Other methods directly control action persistence: TAAC and SDAR learn whether actions should be repeated, TempoRL and FiGAR learn their duration, and multi-frequency control assigns different update rates across action dimensions (Yu et al., 2021; Nie et al., 2025; Biedenkapp et al., 2021; Sharma et al., 2017; Lee et al., 2020). Related approaches also use temporally extended action structures, including exploration-oriented methods such as Generative Planning and DEP-RL, as well as action-interface methods such as RAPS (Zhang et al., 2022; Schumacher et al., 2023; Dalal et al., 2021). Many of these approaches introduce temporal structure by modifying the policy parameterization, temporal decision rule, or action interface. DARE instead operates additively at the executed-action level, leaving the underlying policy architecture and temporal abstraction unchanged.

Step-wise exploration. Step-wise exploration perturbs actions as the agent interacts with the environment. Early methods replaced independent Gaussian noise with autocorrelated processes, while later work introduced autoregressive structure into action generation or the policy itself (Wawrzyński, 2015; Szulc et al., 2020; Korenkevych et al., 2019). Pink

noise takes a different view, showing that the frequency structure of action noise can itself have a strong effect on exploration (Eberhard et al., 2023). ESA instead adapts each sampled action using sinusoidal perturbations and estimated value feedback (Chang and Gao, 2024). These approaches mainly structure the temporal dependence of successive perturbations or adapt individual actions using value feedback. DARE complements this temporal structure by deriving cross-dimensional action modes and allocating exploration according to their response strengths.

3 Preliminaries

3.1 Problem Formulation

We consider a continuous-control Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A} = [-1, 1]^{d_a}$ is the normalized bounded action space with action dimension d_a, $P(\cdot \mid s, a)$ is the distribution of the next state given state s and action a, r is the reward function, and $\gamma \in [0, 1)$ is the discount factor. At state $s_t \in \mathcal{S}$, a policy π samples an action $a_t \sim \pi(\cdot \mid s_t)$.

During online interaction, we consider additive action perturbations applied to the actions generated by the current policy:

$$\tilde{a}_t = \mathrm{clip}(a_t + u_t, -1, 1), \tag{1}$$

where $u_t \in \mathbb{R}^{d_a}$ denotes an external perturbation in the action space. Rather than manually specifying meaningful perturbation directions, DARE aims to discover action-space directions based on the behavioral changes they induce.

3.2 Base Reinforcement Learning Algorithm

Soft Actor-Critic (SAC) is used as the primary off-policy learning algorithm (Haarnoja et al., 2018; 2019). SAC optimizes a stochastic policy under the maximum-entropy objective, encouraging both high return and policy entropy.

The critic is trained using the soft Bellman target

$$y_t = r_t + \gamma \mathbb{E}_{a_{t+1} \sim \pi} \left[Q_{\bar{\theta}}(s_{t+1}, a_{t+1}) - \alpha_{\mathrm{SAC}} \log \pi(a_{t+1} \mid s_{t+1}) \right], \tag{2}$$

with critic objective

$$J_Q(\theta) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}} \left[\frac{1}{2} \left(Q_\theta(s_t, a_t) - y_t \right)^2 \right]. \tag{3}$$

The actor is optimized according to

$$J_\pi = \mathbb{E}_{s_t \sim \mathcal{D},\, a_t \sim \pi} \left[\alpha_{\mathrm{SAC}} \log \pi(a_t \mid s_t) - Q_\theta(s_t, a_t) \right] \tag{4}$$

where $\mathcal{D}$ denotes the replay buffer and α_{SAC} is the SAC entropy-temperature parameter. DARE modifies only the actions used during data collection. The executed action and resulting transition are stored in $\mathcal{D}$. The SAC critic and actor objectives remain unchanged.

4 Methodology

As shown in Figure 1, DARE consists of two stages: basis discovery and online exploration. Basis discovery measures the response to each probe offset by comparing a perturbed rollout with its matched unperturbed rollout (Eqs. (8)–(9)), and extracts the Action-Response

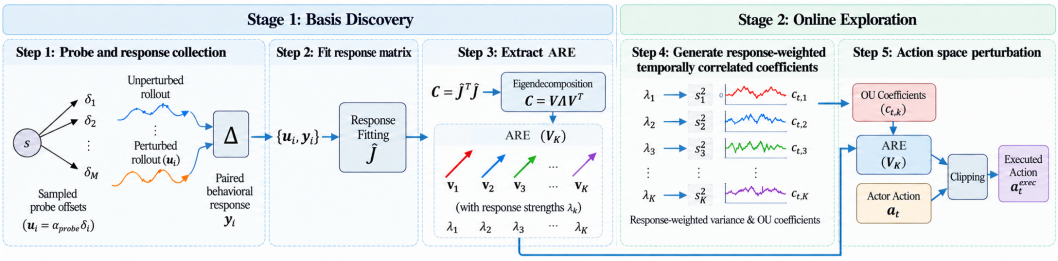


Figure 1: DARE overview. Stage 1 uses paired short rollouts to fit a shared local response matrix and extract ARE directions. Stage 2 applies response-weighted OU perturbations along these directions before action clipping.

Eigenbasis (ARE) from the fitted response matrix. Online exploration then applies temporally correlated perturbations along the discovered directions, weighted by their estimated response strengths.

Here, matching means that the perturbed and unperturbed rollouts start from the same simulator state and use the same random number stream. The policy parameters also remain fixed throughout the paired-rollout collection for a given ARE estimate. Basis discovery further assumes that, within the probe neighborhood, the paired responses at all sampled root states can be approximated by a common local linear mapping from action offsets to behavioral-response vectors, as formalized in Eq. (10).

4.1 Basis Discovery

Two definitions used throughout basis discovery are introduced first: the rollout notation and its segment-level behavioral representation.

$$\tau = \mathrm{Rollout}(s, \pi, H; \omega) = (s_1, \ldots, s_H), \tag{5}$$

where π is the policy, s is the initial state, H is the rollout horizon, and ω denotes the random number stream used during rollout generation.

To obtain a behavioral representation of the rollout, each state is mapped through a feature mapping $\phi : \mathcal{S} \to \mathbb{R}^{d_\phi}$. Stacking these per-state features over the rollout defines the segment-level behavioral representation:

$$z(\tau) = \begin{bmatrix} \phi(s_1) \\ \vdots \\ \phi(s_H) \end{bmatrix} \in \mathbb{R}^{d_z}, \qquad d_z = H d_\phi. \tag{6}$$

The choice of ϕ therefore determines the behavioral representation in which the two executions are compared. Depending on the environment, ϕ may include feature selection, normalization, coordinate transformations, or simply be the identity mapping.

Probe and response collection. For each sampled root state s_g, with $g = 1, \ldots, G_{\mathrm{root}}$, isotropic probe directions are sampled from the normalized action space:

$$\delta_i \sim \mathrm{Unif}\left(\mathbb{S}^{d_a - 1}\right), \qquad u_i = \alpha_{\mathrm{probe}} \delta_i, \qquad i = 1, \ldots, I, \tag{7}$$

Here, $\mathbb{S}^{d_a-1} = \{\delta \in \mathbb{R}^{d_a} : \|\delta\|_2 = 1\}$ and $0 < \alpha_{\mathrm{probe}}$ controls the probe magnitude. For each root state s_g, all I offsets $\{u_i\}_{i=1}^{I}$ are probed.

For an offset u, π_u denotes the perturbed policy obtained by adding u to each action produced by the current policy and clipping the result into the normalized action bounds. For each (s_g, u_i), a perturbed rollout under π_{u_i} and an unperturbed rollout under π are collected using the same probe horizon H_{probe} and the same random stream $\omega_{g,i}$:

$$\tau_{g,i}^{(u)} = \text{Rollout}(s_g, \pi_{u_i}, H_{\text{probe}}; \omega_{g,i}), \qquad \tau_{g,i}^{(0)} = \text{Rollout}(s_g, \pi, H_{\text{probe}}; \omega_{g,i}). \tag{8}$$

Their difference defines the **paired behavioral response**:

$$y_{g,i} = z(\tau_{g,i}^{(u)}) - z(\tau_{g,i}^{(0)}) \in \mathbb{R}^{d_z}. \tag{9}$$

This response measures the change induced by the offset relative to the unperturbed execution, without assuming that the change improves task performance.

Fit 1st-order response matrix. Let $y_g(u)$ represent the expected paired response at root state s_g. For sufficiently small offsets, we assume that the response is locally well approximated by a shared linear mapping across the sampled root states:

$$y_g(u) \approx Ju, \qquad J \in \mathbb{R}^{d_z \times d_a}. \tag{10}$$

This approximation is centered at $u = 0$, for which the paired rollouts give $y_g(0) = 0$. The shared-response assumption allows a single response matrix to be estimated using probe measurements collected from different root states.

The shared matrix J is only a local approximation: the same action offset can produce different responses at different root states because the action–behavior relation may be nonlinear and state dependent. The goal is not to model each response exactly, but to capture the dominant action-response directions shared across the sampled roots.

Then, the matrix can be estimated using ridge regression:

$$\widehat{J} \in \arg\min_{J \in \mathbb{R}^{d_z \times d_a}} \sum_{g=1}^{R} \sum_{i=1}^{I} \|y_{g,i} - Ju_i\|_2^2 + \beta \|J\|_F^2. \tag{11}$$

Here, ($\beta \geq 0$) controls the strength of ridge regularization, which improves stability when the probe directions provide insufficient or poorly conditioned coverage of the action space.

Extract ARE. Given $\widehat{J}$, the next step is to identify unit directions in action space that maximize the magnitude of the predicted behavioral response. Specifically,

$$v_1 \in \arg\max_{\|v\|_2 = 1} \|\widehat{J}v\|_2^2. \tag{12}$$

Since $\|\widehat{J}v\|_2^2 = v^\top \widehat{J}^\top \widehat{J} v$, maximizing the predicted response over unit directions reduces to an eigenvalue problem of $\widehat{J}^\top \widehat{J}$. The resulting directions are therefore given by the eigenvectors of $\widehat{J}^\top \widehat{J}$. A full derivation is provided in Appendix B.1.

$$C = \widehat{J}^\top \widehat{J}, \qquad Cv_k = \lambda_k v_k, \qquad \lambda_1 \geq \cdots \geq \lambda_{d_a} \geq 0. \tag{13}$$

The eigenvectors associated with positive eigenvalues define up to $\min(d_z, d_a)$ orthogonal action-space response modes, with each λ_k giving the predicted squared response magnitude along v_k. The properties of these modes are derived in Appendix B.1.

Modes with eigenvalues exceeding a small numerical tolerance ε_{eig} are retained. Let

$$K = |\{k : \lambda_k > \varepsilon_{\text{eig}}\}|, \qquad V_K = [v_1, \ldots, v_K] \in \mathbb{R}^{d_a \times K}. \tag{14}$$

The retained modes form the ARE for online exploration. Unless otherwise stated, DAB is discovered once before online learning and kept fixed throughout training to isolate the effect of the discovered basis during online exploration.

4.2 Online Exploration

Using the ARE modes V_K and corresponding response strengths $\{\lambda_k\}_{k=1}^K$ from Eq. (14), DARE constructs a temporally correlated exploration process in the original action space.

Response-weighted temporally correlated coefficients. A hyperparameter $\mathcal{E}_b > 0$ controls the overall exploration energy allocated across the retained ARE modes.

For each retained ARE mode v_k, let $c_{t,k}$ denote its exploration coefficient and s_k^2 its stationary variance. This variance is allocated in proportion to the mode response strength:

$$s_k^2 = \mathcal{E}_b \frac{\lambda_k^\alpha}{\sum_{j=1}^K \lambda_j^\alpha}, \qquad k = 1, \dots, K, \tag{15}$$

where $0 \le \alpha \le 1$ controls the degree of response-based weighting. When $\alpha = 0$, all modes receive equal variance, whereas $\alpha = 1$ leads to larger variance for modes with stronger responses, and vice versa.

To introduce temporal correlation, each coefficient evolves according to an independent stationary Ornstein–Uhlenbeck process and determines the instantaneous perturbation magnitude along its corresponding mode::

$$c_{t+1,k} = \eta c_{t,k} + \sqrt{1-\eta^2}\, s_k \xi_{t,k}, \qquad \xi_{t,k} \sim \mathcal{N}(0,1), \tag{16}$$

where $\eta = \exp(-1/H_{\mathrm{OU}})$ controls the temporal persistence of the perturbation and the hyperparameter H_{OU} determines the correlation timescale; $\xi_{t,k}$ is an independent standard Gaussian innovation. The coefficient is initialized from its stationary distribution:

$$c_{0,k} \sim \mathcal{N}(0, s_k^2). \tag{17}$$

Thus, s_k, as the stationary standard deviation of $c_{t,k}$, controls the perturbation scale along mode v_k, while H_{OU} controls how long the perturbation persists.

Action-space perturbation. At each environment step, the mode coefficients $\{c_{t,k}\}_{k=1}^K$ are collected into c_t. These coefficients scale the corresponding ARE modes $\{v_k\}_{k=1}^K$ and are combined to form a pre-clipping exploration bias in the original action space:

$$c_t = [c_{t,1}, \dots, c_{t,K}]^\top, \qquad b_t = V_K c_t = \sum_{k=1}^K c_{t,k}\ v_k. \tag{18}$$

The executed action is

$$a_t^{\mathrm{exec}} = \mathrm{clip}\left(a_t^{\mathrm{actor}} + b_t, -1, 1\right). \tag{19}$$

The replay buffer stores the executed action a_t^{exec}, as the observed transition is generated by this action. Under the stationary coefficient distribution, the expected squared magnitude of the pre-clipping bias equals the preset exploration energy ($\mathbb{E}[\|b_t\|_2^2] = \mathcal{E}_b$), while its allocation across ARE modes follows their estimated behavioral response strengths.

5 Experiments

The experiments are designed to answer four questions: under what task conditions DARE is most effective; whether its benefits transfer across different base learning algorithms; which of its components contribute to its performance; and whether DARE can complement other exploration methods rather than being used only as a standalone exploration strategy.

5.1 Experimental Setup

Experiments are conducted in ManiSkill 3.0.1 (Tao et al., 2025) on several robot control tasks with sparse rewards and no expert data. All tasks use their default step horizon. SAC (Haarnoja et al., 2018; 2019) is the primary learning algorithm, with PPO (Schulman et al., 2017) and TD3 (Fujimoto et al., 2018) used for cross-algorithm evaluation. Within each task–learning algorithm configuration, all methods share policy initialization, reset sequences, learning algorithm hyperparameters, and online interaction budgets. See details in Appendix A.2.

The main comparison includes the base learning algorithm (**SAC**) and representative exploration mechanisms, including unstructured action perturbations (**Segment-wise IID**), temporally correlated noise (**OU Noise** and **Pink Noise** (Eberhard et al., 2023)), state-dependent exploration (**gSDE** (Raffin et al., 2022)), and extremum-seeking action selection (**ESA** (Chang and Gao, 2024)). Lattice (Chiappa et al., 2023) and DEP-RL (Schumacher et al., 2023) are also relevant structured-exploration methods, but are not included in the main comparison. The main comparison keeps the learned policy and its action-producing mechanism unchanged, while varying the exploration mechanism used with that policy. Lattice instead perturbs the policy latent state directly, while DEP-RL intermittently generates actions using a separate controller. Their performance would therefore reflect changes beyond the exploration mechanism considered in the main comparison.

Results are averaged over five or three matched seeds, each with 5,120 trajectories and 10,240 policy updates. Exploration is evaluated by cumulative reaches of five task-specific ordered milestones L_0–L_4 (Appendix A.1). DARE probing interactions are recorded separately from the online training budget.

5.2 Main Results

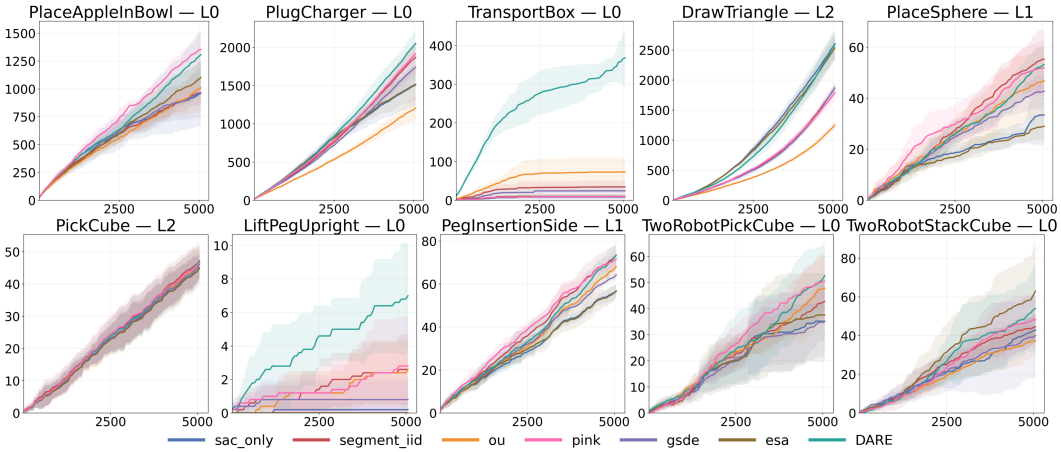


Figure 2: Accumulative milestone reaching curves during online exploration for the most discriminative milestone in each task; The x-axis denotes the cumulative number of exploration trajectories collected during training, and the y-axis shows how often exploration visits the corresponding task-progress region; full curves in Appendix C.2.

Across the evaluated tasks in Figure 2, DARE shows its clearest advantage on TransportBox, while LiftPegUpright also favors DARE despite the relatively small number of reaches. On the remaining tasks, DARE is generally close to the strongest baseline, with only modest differences.

DARE's clearest gains emerge on tasks where competing methods make limited progress. On TransportBox, most baselines plateau early, whereas DARE continues to improve throughout training. LiftPegUpright shows a similar pattern, with DARE making greater progress while most alternatives remain near low performance levels. On PickCube, by contrast, all methods follow nearly identical learning curves. These results highlight DARE's ability to sustain learning in settings where existing exploration strategies struggle, although it does not achieve the highest performance on every task.

5.3 Cross-Algorithm Evaluation

To test whether the gains are SAC-specific, TD3 and PPO are also evaluated on the task TransportBox, comparing base learning algorithm, base learning algorithm+OU, and base learning algorithm+DARE under matched settings. Configurations are detailed in Appendices A.2 and A.3.

Table 1: Cumulative milestone reaches on `UnitreeG1TransportBox-v1` under TD3 and PPO (mean ± sample s.d. over five matched seeds). Boldface indicates the larger mean or the smaller sample s.d.

Learner	Method	L_0	L_1
TD3	DARE	$\mathbf{4507.40 \pm 598.46}$	$\mathbf{24.20 \pm 35.94}$
	Base	3280.40 ± 1358.23	1.20 ± 2.17
	OU	3558.80 ± 1709.78	0.00 ± 0.00
PPO	DARE	$\mathbf{4289.80 \pm 218.05}$	4.00 ± 3.46
	Base	3560.40 ± 896.59	0.40 ± 0.55
	OU	3881.20 ± 610.92	3.80 ± 3.27

TransportBox, where DARE shows its strongest advantage under SAC, remains the most favorable task after switching to TD3 and PPO. DARE continues to achieve the strongest results with significantly smaller standard deviations under both learning algorithms, indicating that its benefit transfers across algorithms. However, the margin is smaller than under SAC, with the base learning algorithms already approaching the lower end of DARE performance at L_0. This suggests that DARE generalizes across learning algorithms, while the magnitude of its advantage remains learning algorithm-dependent. Full L-stages reaching curves and seed-level results are provided in Appendix C.2.

5.4 Ablation Studies

Table 2: Ablation results on Allegro In-Hand Rotation (mean ± s.d. over three seeds).

Method	L_0	L_1	L_2	L_3	L_4
Raw DAB-7D (no OU, hold-8)	4829.7 ± 67.6	2712.7 ± 327.4	705.7 ± 166.1	168.0 ± 54.3	53.7 ± 21.2
DARE Uniform-7D	$\mathbf{4833.7 \pm 49.2}$	$\mathbf{2900.7 \pm 136.8}$	$\mathbf{964.7 \pm 192.6}$	225.7 ± 70.1	56.0 ± 17.3
DARE Spectral General-73D	4815.3 ± 68.0	2595.7 ± 230.6	706.3 ± 87.9	157.7 ± 20.0	38.0 ± 6.9
DARE Spectral ObjectPose-7D	4721.3 ± 62.7	2694.7 ± 150.3	907.0 ± 118.5	$\mathbf{258.7 \pm 20.0}$	$\mathbf{71.0 \pm 3.6}$

Ablations are conducted on Allegro because its milestone reaches span from L_0 to L_4, allowing changes that may be weak at early milestones to become visible at deeper ones. The study examines the DARE formulation, response-weighted allocation, and the response representation ϕ, with full configurations provided in Appendix A.4.

Raw ARE and Uniform DARE perform similarly overall, with small gains and losses across milestones. Response weighting produces a consistent improvement, particularly at deeper milestones. The response representation is also important: replacing General-73D with ObjectPose-7D improves all milestones, with the gap increasing toward L_4. These results highlight both response-weighted allocation and the choice of behavioral features as important components of DARE.

Sensitivity to the exploration-energy, temporal-correlation, and mode-number settings is reported in Appendix C.1.

Table 3: Compatibility of DARE with complementary exploration methods on Allegro.

Method	L_0	L_1	L_2	L_3	L_4
SDAR	4792.3 ± 76.6	2752.0 ± 605.9	926.3 ± 239.8	237.3 ± 74.7	65.0 ± 29.2
SDAR + DARE	$\mathbf{4853.0 \pm 13.9}$	$\mathbf{3077.3 \pm 42.0}$	$\mathbf{1110.3 \pm 79.0}$	$\mathbf{297.7 \pm 47.9}$	$\mathbf{74.7 \pm 4.0}$
VIME	4773.3 ± 81.5	2873.3 ± 208.6	863.3 ± 359.7	201.7 ± 104.8	48.3 ± 11.6
VIME + DARE	$\mathbf{4805.7 \pm 24.7}$	2698.7 ± 327.6	$\mathbf{873.7 \pm 235.1}$	$\mathbf{220.7 \pm 84.7}$	$\mathbf{61.7 \pm 12.4}$

5.5 Complementarity with Other Exploration Mechanisms

DARE acts only at the executed-action level and can therefore be combined with exploration mechanisms operating through other structures or objectives. Table 3 provides representative combination examples. Adding DARE to SDAR produces a modest but consistent improvement across all miletones, with higher mean counts and generally lower variability across seeds. While adding DARE to VIME also improves mean on most of the milestones. This illustrates that response-weighted action perturbations can provide complementary gains on top of some other exploration mechanism.

6 Discussion and Limitations

In this paper, we presented DARE, an exploration method that uses local behavioral responses to identify action directions. From paired short rollouts, DARE constructs action response bases and uses them to guide temporally correlated exploration, without requiring expert actions, dense rewards, or externally specified directions. The central idea is to organize exploration around the behavioral effects of actions, allowing exploratory perturbations to follow directions grounded in the system's response.

Results on several Maniskill tasks show the strongest empirical evidence, where DARE makes sustained progress while competing methods plateau or improve only slowly. These observations are consistent with the motivation behind DARE: maintaining exploration along behaviorally meaningful directions can be valuable when existing exploration strategies make limited progress. DARE does not lead on every task, and the learning curves alone do not establish which task properties account for the differences. They nevertheless demonstrate that response-guided exploration can yield substantial gains in settings where the alternatives struggle.

As for future work, first, DARE discovers its response basis once and keeps it fixed throughout training. It's worthwhile investigating how an adaptive update mechanism could refresh the basis when the observed responses depart from those represented by the initial basis. Second, DARE uses a fixed OU process to generate exploration coefficients. Adapting the persistence and weighting of these coefficients could allow exploration to respond to changing behavioral demands.

AI use statement

Generative AI tools were used to assist with several parts of this work. They assisted with literature retrieval and research survey, including identifying related work and supporting background investigation. They were also used to assist with parts of the experimental design, including the design of milestone definitions for some tasks, the implementation of experimental code, and the organization and processing of experimental results. Generative AI tools also assisted with the proof presented in the section "Policy Improvement Does Not Guarantee Beneficial ARE Rediscovery." In addition, they assisted with generating the method overview figure from author-provided descriptions, as well as drafting and polishing parts of the manuscript.

All AI-assisted materials were reviewed, tested where applicable, and revised by the authors. The authors take responsibility for the final text, code, figures, experimental design choices, mathematical arguments, claims, and conclusions in this work.

References

Susan Amin, Maziar Gomrokchi, Hossein Aboutalebi, Harsh Satija, and Doina Precup. Locally persistent exploration in continuous control tasks with sparse rewards. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 275–285. PMLR, 2021.

Nir Baram, Guy Tennenholtz, and Shie Mannor. Action redundancy in reinforcement learning. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 376–385. PMLR, 2021. URL `https://proceedings.mlr.press/v161/baram21a.html`.

André Biedenkapp, Raghu Rajan, Frank Hutter, and Marius Lindauer. TempoRL: Learning when to act. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 914–924. PMLR, 2021.

Ya-Chien Chang and Sicun Gao. Extremum-seeking action selection for accelerating policy optimization. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5141–5147. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10610197.

Alberto Silvio Chiappa, Alessandro Marin Vargas, Ann Huang, and Alexander Mathis. Latent exploration for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2466.

Murtaza Dalal, Deepak Pathak, and Ruslan Salakhutdinov. Accelerating robotic reinforcement learning via parameterized action primitives. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Onno Eberhard, Jakob Hollenstein, Cristina Pinneri, and Georg Martius. Pink noise is all you need: Colored noise exploration in deep reinforcement learning. In *International Conference on Learning Representations*, 2023.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR, 2018. URL `https://proceedings.mlr.press/v80/fujimoto18a.html`.

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*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 2018.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2019.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. VIME: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, volume 29, pages 1109–1117, 2016.

Dmytro Korenkevych, A. Rupam Mahmood, Gautham Vasan, and James Bergstra. Autoregressive policies for continuous control deep reinforcement learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 2754–2762, 2019. doi: 10.24963/ijcai.2019/382.

Jongmin Lee, Byung-Jun Lee, and Kee-Eung Kim. Reinforcement learning for control with multiple frequencies. In *Advances in Neural Information Processing Systems*, volume 33, pages 3254–3264, 2020.

Ge Li, Hongyi Zhou, Dominik Roth, Serge Thilges, Fabian Otto, Rudolf Lioutikov, and Gerhard Neumann. Open the black box: Step-based policy updates for temporally-correlated episodic reinforcement learning. In *International Conference on Learning Representations*, 2024.

Ge Li, Dong Tian, Hongyi Zhou, Xinkai Jiang, Rudolf Lioutikov, and Gerhard Neumann. TOP-ERL: Transformer-based off-policy episodic reinforcement learning. In *International Conference on Learning Representations*, 2025.

Buqing Nie, Yangqing Fu, and Yue Gao. Select before act: Spatially decoupled action repetition for continuous control. In *International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=PDgZ3rvqHn`.

Fabian Otto, Onur Celik, Hongyi Zhou, Hanna Ziesche, Vien Anh Ngo, and Gerhard Neumann. Deep black-box reinforcement learning with movement primitives. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1244–1265. PMLR, 2023.

Silviu Pitis, Harris Chan, Stephen Zhao, Bradly Stadie, and Jimmy Ba. Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7750–7761. PMLR, 2020.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *International Conference on Learning Representations*, 2018.

Vitchyr Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7783–7792. PMLR, 2020.

Antonin Raffin, Jens Kober, and Freek Stulp. Smooth exploration for robotic reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 1634–1644. PMLR, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL `https://arxiv.org/abs/1707.06347`.

Pierre Schumacher, Daniel F. B. Haeufle, Dieter Büchler, Syn Schmitt, and Georg Martius. DEP-RL: Embodied exploration for reinforcement learning in overactuated and musculoskeletal systems. In *International Conference on Learning Representations*, 2023.

Lukas Schäfer, Filippos Christianos, Josiah P. Hanna, and Stefano V. Albrecht. Decoupled reinforcement learning to stabilise intrinsically-motivated exploration. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2022.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020.

Sahil Sharma, Aravind S. Lakshminarayanan, and Balaraman Ravindran. Learning to repeat: Fine grained action repetition for deep reinforcement learning. In *International Conference on Learning Representations*, 2017.

Marcin Szulc, Jakub Łyskawa, and Paweł Wawrzyński. A framework for reinforcement learning with autocorrelated actions. *arXiv preprint arXiv:2009.04777*, 2020.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. ManiSkill3: GPU parallelized robotics simulation and rendering for generalizable embodied AI. *Robotics: Science and Systems*, 2025.

Herke van Hoof, Daniel Tanneberg, and Jan Peters. Generalized exploration in policy search. *Machine Learning*, 106(9–10):1705–1724, 2017. doi: 10.1007/s10994-017-5657-1.

Paweł Wawrzyński. Control policy with autocorrelated noise in reinforcement learning for robotics. *International Journal of Machine Learning and Computing*, 5(2):91–95, 2015. doi: 10.7763/IJMLC.2015.V5.489.

Haonan Yu, Wei Xu, and Haichao Zhang. TAAC: Temporally abstract actor-critic for continuous control. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Xuehui Yu, Mhairi Dunion, Xin Li, and Stefano V. Albrecht. Skill-aware mutual information optimisation for generalisation in reinforcement learning. In *Conference on Neural Information Processing Systems*, 2024.

Haichao Zhang, Wei Xu, and Haonan Yu. Generative planning for temporally coordinated exploration in reinforcement learning. In *International Conference on Learning Representations*, 2022.

Yijie Zhang and Herke Van Hoof. Deep coherent exploration for continuous control. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12567–12577. PMLR, 2021.

A Experimental Details

A.1 Tasks, Rewards, and Metrics

Table 4: Tasks and response representations.

Agent	d_o	d_a	H	d_ϕ	Response features
In-Hand Rotation `RotateSingleObjectInHandLevel0-v1`					
Allegro right hand with tactile sensors	102	16	300	7	Object position (3) and quaternion (4): ObjectPose.
TransportBox `UnitreeG1TransportBox-v1`					
G1 upper body with head camera	77	25	100	77	qpos (25), qvel (25), two TCP poses (14), box pose (7), box velocities (6).
PlaceAppleInBowl `UnitreeG1PlaceAppleInBowl-v1`					
G1 upper body with head camera	74	25	100	77	qpos (25), qvel (25), two TCP poses (14), apple pose (7), apple velocities (6).
StackPyramid `StackPyramid-v1`					
Panda	64	8	250	64	qpos (9), qvel (9), TCP pose (7), and three cube pose/velocity blocks (13 each).
DrawTriangle `DrawTriangle-v1`					
PandaStick	49	7	300	21	qpos (7), qvel (7), TCP pose (7).
PlugCharger `PlugCharger-v1`					
Panda with wrist camera	46	8	200	46	Native policy/state vector, in adapter order.
PlaceSphere `PlaceSphere-v1`					
Panda	39	8	50	39	qpos/qvel (18), TCP/object poses (14), bin position (3), TCP-to-object vector (3), grasp flag (1).
PickCube `PickCube-v1`					
Panda	42	8	50	42	qpos/qvel (18), TCP/object poses (14), goal position (3), TCP-to-object and object-to-goal vectors (6), grasp flag (1).
LiftPegUpright `LiftPegUpright-v1`					
Panda	32	8	50	32	qpos/qvel (18), TCP/peg poses (14).
PegInsertionSide `PegInsertionSide-v1`					
Panda with wrist camera	43	8	100	43	qpos/qvel (18), TCP/peg/hole poses (21), peg half-size (3), hole radius (1).
TwoRobotStackCube `TwoRobotStackCube-v1`					
Two Pandas with wrist cameras	76	16	100	76	Both robots' qpos/qvel (36), two TCP poses (14), two cube poses (14), goal position (3), two TCP-to-cube and inter-cube vectors (9).
TwoRobotPickCube `TwoRobotPickCube-v1`					
Two Pandas with wrist cameras	66	16	100	66	Both robots' qpos/qvel (36), two TCP poses (14), cube pose (7), two TCP-to-cube and cube-to-goal vectors (9).

All tasks use ManiSkill 3.0.1, GPU PhysX, state observations, and `pd_joint_delta_pos` actions in $[-1, 1]^{d_a}$; control/simulation frequencies are 20/100 Hz. d_o, d_a, and d_ϕ denote policy-observation, action, and per-step response dimensions; H is the horizon in control steps. qpos/qvel denote joint positions/velocities; TCP denotes tool center point. Poses contain position and quaternion; object/base velocities contain linear and angular components. The six added tasks use their native flattened state as the response. Response features are used for DARE basis fitting without enlarging network inputs. Both two-robot tasks use one joint policy with $8 + 8 = 16$ action coordinates.

Table 5: Milestone conditions. Grouped levels list thresholds in order.

Task	Level	Condition
In-Hand Rotation	L_0	$\theta \geq \pi/6$.
	L_1	$\theta \geq \pi/2$.
	L_2	$\theta \geq \pi$.
	L_3	$\theta \geq 2\pi$.
	L_4	Native success: $\theta > 4\pi$.
TransportBox	L_0	Face the target table.
	L_1	Both hands contact the box.
	L_2	Native `box_grasped`.
	L_3	Box in the target-table xy region.
	L_4	Native success.
PlaceAppleInBowl	L_0	TCP-to-apple distance ≤ 0.10 m.
	L_1	Native apple grasp.
	L_2	Maintain grasp; lift apple ≥ 0.05 m above reset height.
	L_3	Maintain grasp; apple within 0.025 m of a point 0.15 m above the bowl.
	L_4	Native success.
StackPyramid	L_0	$d_{xy}(A, B) \leq 0.12$ m.
	L_1	$d_{xy}(A, B) \leq R$.
	L_2	$d_{xy}(C, A), d_{xy}(C, B) \leq 0.12$ m.
	L_3	Native stacking geometry, including the $d_{xy} \leq R$ and height-gap tests.
	L_4	Native success.
DrawTriangle	L_0	First ink point deposited.
	L_1	First valid visible ink point.
	L_2	At least 16 of 153 reference points covered.
	L_3	At least 77 of 153 reference points covered.
	L_4	All 153 reference points covered; every visible dot valid.
PlugCharger	L_0	TCP-to-charger-base grasp-point distance ≤ 0.08 m.
	L_1	Native Panda `is_grasping(charger)`.
	L_2	Maintain grasp; lift charger ≥ 0.02 m above reset height.
	L_3	Maintain grasp; goal distance ≤ 0.015 m and angle error ≤ 0.35 rad.
	L_4	Native success: distance ≤ 0.005 m and angle error ≤ 0.2 rad.
PlaceSphere	L_0–L_2	Sphere-center distance to the bin's target center ≤ 0.12, 0.06, 0.025 m, respectively.
	L_3	Native placement: horizontal and vertical errors each ≤ 0.005 m.
	L_4	Native success: placed, sphere static, and not grasped.
PickCube	L_0–L_2	Cube-to-goal distance ≤ 0.15, 0.08, 0.04 m, respectively.
	L_3	Native placement: cube-to-goal distance ≤ 0.025 m.
	L_4	Native success: placed and robot static.
LiftPegUpright	L_0–L_3	Peg long-axis elevation above the horizontal plane $\geq 15°$, 30°, 60°, 80°, respectively.
	L_4	Long-axis deviation from vertical < 0.08 rad and $\lvert z_{\text{peg}} - \ell \rvert < 0.005$ m; ℓ is the peg half-length.
PegInsertionSide	L_0–L_3	Peg-tip distance to the native insertion-success region ≤ 0.30, 0.15, 0.06, 0.02 m, respectively.
	L_4	Native insertion: in hole coordinates, $x \geq -0.015$ m, $\lvert y \rvert \leq r_h$, and $\lvert z \rvert \leq r_h$.

Continued on next page

Table 5 (continued)

Task	Level	Condition
TwoRobot StackCube	L_0	Base cube B within 0.18 m of the target in xy.
	L_1	Native placement of B: target xy distance < 0.06 m.
	L_2	B currently placed; cube A within 0.10 m of the stacking center above B.
	L_3	B currently placed; native A-on-B geometry: xy offset $\leq R_2$, height-gap error ≤ 0.005 m.
	L_4	Native success: B placed, A stacked, and the corresponding grippers release both cubes.
TwoRobot PickCube	L_0	Cube world $y \geq -0.05$ m.
	L_1	Cube world $y \geq 0$.
	L_2	Cube-to-goal distance ≤ 0.08 m.
	L_3	Native placement: cube-to-goal distance ≤ 0.025 m.
	L_4	Native success: placed and right robot static.

All levels require the complete milestone prefix, attained earlier or on the same step; premature conditions are not cached. Each level is counted and rewarded once per trajectory; latches reset between trajectories. Distances are three-dimensional unless marked xy. θ is the native rotation-angle statistic. For StackPyramid, $R = \|2\,\texttt{cube_half_size}_{xy}\|_2 + 0.005$ m (≈ 0.06157 m); no grasp/lift/release route is required. DrawTriangle validity and coverage use planar distance < 0.025 m.
For PlaceSphere, the target is the bin position plus sphere radius and bin-bottom half-thickness in z; sphere-static thresholds are 0.01 m/s and 0.5 rad/s. LiftPegUpright uses $\arcsin|R_{zx}|$ for long-axis elevation and corrects the installed native Euler-axis error in success and termination. For PegInsertionSide, r_h is the actual hole radius and the approach distance is $\|(\max(-0.015 - x, 0), \max(|y| - r_h, 0), \max(|z| - r_h, 0))\|_2$. For TwoRobotStackCube, the stacking center is $p_B + (0, 0, 2h_z)$ and $R_2 = \|h_{xy}\|_2 + 0.005$ m (≈ 0.03328 m), where h is cube half-size. PickCube and TwoRobotPickCube use the native robot-static threshold 0.2.

A.2 Shared Training Settings

Table 6: Training and reporting protocol. The six added tasks use the separately stated collection budget.

Setting	Value
Seeds / environments	Original six-task protocol: seeds 0–2; added tasks: seeds 1–5. All use 64 parallel environments.
Collection budget	10 rounds × 512 trajectories = 5,120 per seed. Updates follow each batch of 64 trajectories.
Collection policy	Networks fixed within each collection batch; no random-action warm-up.
Network / optimizer	Two 512-unit ReLU hidden layers; Adam, learning rate 3×10^{-4}.
Discount / normalization	$\gamma = 0.97$; no running observation or reward normalization.
SAC/TD3 replay	Capacity $BH + 64$ transitions, where B is the per-seed trajectory budget; uniform sampling with replacement; minibatch size 1,024.
PPO storage	Current on-policy rollout only; no replay buffer.
Bootstrapping	Both task termination and timeouts stop bootstrapping.
Milestone reward	$r_t = \sum_{i=0}^{4} w_i(m_{i,t} - m_{i,t-1})$, $(w_0, \ldots, w_4) = (0.1, 0.2, 0.3, 0.5, 1.0)$; $m_{i,t}$ records whether L_i has been attained by step t.
Other rewards	Native rewards logged separately, not added; no intrinsic bonus except in the VIME combination.
Reported metric	Cumulative milestone-reaching trajectory counts during active exploration; per-step records and summaries every 64 trajectories; mean ± sample standard deviation across matched seeds.
Evaluation / checkpoint	No separate evaluation during collection; final scheduled checkpoint, without validation-based selection.

All tasks comparison has 10,240 SAC updates per optimizer per seed.

Table 7: Learner hyperparameters.

Setting	SAC	TD3	PPO
Policy	Gaussian + tanh	Deterministic tanh	Gaussian + tanh
Value model	Twin $Q(o, a)$	Twin $Q(o, a)$	$V(o)$
Input LayerNorm	No	No	Yes
Policy log-std bounds	$[-5, 2]$	—	$[-5, 2]$
Updates per collection batch	128 critic, actor, and temperature steps	128 critic; 64 actor steps	4 epochs, up to 32 minibatches per epoch
Target mixing τ	0.005	0.005	—
Target updates	Every gradient step	Every actor step	—
Entropy temperature	Initial 0.2; learned	—	—
Target entropy	$-d_a$	—	—
Target-smoothing std / clip	—	0.2 / 0.5	—
GAE λ / ratio clip	—	—	0.95 / 0.2
Entropy / value-loss coefficients	—	—	0 / 0.5
Gradient-norm clipping	None	None	0.5

SAC constrains $\log \alpha_{\mathrm{SAC}} \in [\log 0.01, 5]$. PPO clips gradients separately for each optimized network and uses only active rollout transitions. Input LayerNorm is separate from running observation normalization.

A.3 Exploration and DARE Discovery

Table 8: Exploration configurations.

Method	Configuration
SAC-only	Base policy sampling; no external add-on.
Segment-wise IID	Isotropic Gaussian direction; radius is the norm of six independent draws from $\{2, 3, 4\}/\sqrt{58}$. Bias held for 8 control steps.
OU	Stationary isotropic Ornstein–Uhlenbeck process; e-folding time $H_{\mathrm{OU}} = 8$ control steps.
gSDE-style	Root-mean-square-normalized projection of 512D actor features; resampled every 8 steps and added after the SAC tanh action.
ESA	Per-episode sinusoidal critic-query dither: amplitude 0.2, frequency 2–4 Hz, high-pass cutoff 0.5 Hz, gain $0.1 \to 0.01$. The integrated correction is executed with action bounds enforced.
DARE	All $K = d_a$ response modes; spectral variances $s_k^2 = \mathcal{E}_b \lambda_k / \sum_{j=1}^{K} \lambda_j$. The OU e-folding horizon H_{OU} is task-specific: Allegro (8), TransportBox (8), PlaceAppleInBowl (4), StackPyramid (2), DrawTriangle (1), and PlugCharger (1).

Segment-wise IID, OU, Pink, gSDE-style, and DARE use nominal pre-clipping external-bias energy $\mathcal{E}_b = 1$; base-policy stochasticity is excluded. Pink normalization applies to the full generated trace, not a possibly truncated active prefix. ESA is not energy-matched. DARE uses eigenvalues of $\widehat{J}^\top \widehat{J}$ and stationary coefficient variances s_k^2. All add-ons retain the SAC update equations. The listed DARE H_{OU} values are unchanged; no additional-task DARE horizons are selected here.

Table 9: DARE discovery settings and configured budgets.

Setting	Value
Root states N	64 per seed: 48 fitting and 16 held-out roots.
Probe directions P	d_a normalized Gaussian directions; the current batch runner shares one sampled set across roots within each seed.
Probe horizon / radius	$H_p = 8$ control steps / $\delta = 0.05$; constant offset within a probe.
Paired rollouts	Matched root simulator state and policy innovations; initial learner frozen during discovery; one extra zero–zero pair per root.
Response construction	Align quaternion signs, concatenate paired feature differences over H_p steps, and scale response blocks using fitting roots.
Ridge coefficient / modes	$\beta = 10^{-3}$; retain all $K = d_a$ modes.
Basis use	Fixed during training; learner-specific injection coordinates (Table 10).
Probe data	Reward-blind; excluded from learner samples and accounted separately from training collection.

Task	Stacked dimension d_z	Configured transitions
In-Hand Rotation	56	17,408
TransportBox	616	26,624
PlaceAppleInBowl	616	26,624
StackPyramid	512	9,216
DrawTriangle	168	8,192
PlugCharger	368	9,216
PlaceSphere	312	9,216
PickCube	336	9,216
LiftPegUpright	256	9,216
PegInsertionSide	344	9,216
TwoRobotStackCube	608	17,408
TwoRobotPickCube	528	17,408

$d_z = H_p d_\phi$; per-seed probe budgets are $2NH_p(P+1)$, including the zero–zero pair. These are configuration-derived budgets, not measured completed-run costs; for the six added tasks, they apply if DARE is enabled. For a response block of width D_b, the coordinate weight is $(s_b\sqrt{D_b})^{-1}$ when $s_b > \max(10n_b, 10^{-12})$, otherwise zero; s_b, n_b are fitting-root signal/noise RMS values. Historical reused artifacts retain their original representations, direction sampling, scaling, and recorded interaction costs.

A.4 Additional Experiments

Table 10: Learner-specific integration and calibration.

Setting	Specification
SAC / TD3	$a_t^{\text{exec}} = \text{clip}(a_t^{\text{base}} + b_t, -1, 1)$, with sampled SAC or deterministic TD3 base action. Replay stores the executed action; actor/target computations exclude b_t. SAC entropy and TD3 target smoothing are unchanged; TD3+DARE adds no IID collection noise.
PPO action	Sample $z_t \sim \pi^z_{\theta_{\text{old}}}(\cdot \mid o_t)$ from the actor's diagonal Gaussian, then execute $a_t^{\text{exec}} = \text{clip}(\tanh(z_t) + b_t, -1, 1)$. Actor and value inputs remain o_t.
PPO update	Store the original z_t and its Gaussian log probability. Use $\omega_t(\theta) = \pi^z_\theta(z_t \mid o_t)/\pi^z_{\theta_{\text{old}}}(z_t \mid o_t)$, reusing the same recorded z_t across epochs. This is a latent-action likelihood ratio; the executed action and exploration bias do not enter the log-probability calculation.
Basis fitting	Frozen initial learner; post-tanh action offsets for SAC/TD3, pre-tanh mean shifts for PPO.
TD3/PPO calibration	Fit a scalar gain per task, seed, and noise-bearing arm to initial mean squared executed-action displacement ≈ 1: 64 shared reset observations, 256 samples each. PPO action pairs share Gaussian innovations. Gains are reward-free and fixed during training.

Training follows the original six-task budget in Table 6 and Table 7. Calibration matches initial executed displacement, not nominal pre-clipping energy or energy throughout training.

Table 11: Allegro ablation variants.

Variant	Response	Temporal process	Allocation
DARE (hold-8)	ObjectPose (7D)	Discrete coefficients held for 8 steps	Scaled three-level draws
DARE Uniform	ObjectPose (7D)	OU, $H_{\text{OU}} = 8$	$s_k^2 = 1/16$
DARE Spectral General	General73 (73D)	OU, $H_{\text{OU}} = 8$	$s_k^2 = \lambda_k / \sum_j \lambda_j$
DARE Spectral ObjectPose	ObjectPose (7D)	OU, $H_{\text{OU}} = 8$	$s_k^2 = \lambda_k / \sum_j \lambda_j$

All variants use SAC and the original shared training/discovery settings, except for six matched seeds (0–5). Each seed's discovery data are shared across variants; the 102D network observation is unchanged. General73 contains qpos (16), qvel (16), fingertip poses (28), object pose (7), and object linear/angular velocities (6). The hold-8 variant independently draws $\xi_k \sim \text{Unif}\{\pm 0.2, \pm 0.3, \pm 0.4\}$ per segment, with $c_k = \xi_k \sqrt{\mathcal{E}_b / [K(0.2^2 + 0.3^2 + 0.4^2)/3]}$. Here $K = 16$, $\mathcal{E}_b = 1$; all variants use ordinary action clipping without an extra limiter.

Table 12: Exploration-method combinations.

Task	Comparison	Integration
TransportBox	SDAR vs. SDAR+DARE	Native action/repeat selection followed by external DARE.
In-Hand Rotation	SAC+VIME vs. SAC+VIME+DARE	Retain the VIME intrinsic bonus and add a fixed spectral DARE bias.

Both comparisons follow the original training protocol. Base learner and method-specific updates remain unchanged; DARE adds no optimizer. Executed bounded actions are stored in replay/history. Bases remain fixed; probe data are excluded from learner samples. VIME final-policy evaluation is stochastic with DARE and VIME disabled, reported separately from active-exploration collection counts.

B Theoretical Properties of RW-DAB

B.1 Spectral Properties of ARE Modes

Recall from Eq. (13) that

$$C = \widehat{J}^\top \widehat{J}, \qquad Cv_k = \lambda_k v_k,$$

where the eigenvectors $\{v_k\}_{k=1}^{d_a}$ are chosen to be orthonormal and $\lambda_1 \geq \cdots \geq \lambda_{d_a} \geq 0$.

Proposition 1 (Optimality and orthogonality of response modes)**.** *For every ARE mode v_k,*

$$\|\widehat{J}v_k\|_2^2 = \lambda_k.$$

Moreover,

$$v_1 \in \arg\max_{\|v\|_2=1} \|\widehat{J}v\|_2^2,$$

and, more generally, v_k maximizes the predicted response among unit directions orthogonal to $v_1, \ldots, v_{k-1}$.

For two distinct modes v_j and v_k,

$$(\widehat{J}v_j)^\top (\widehat{J}v_k) = 0.$$

Proof. For any unit vector v,

$$\|\widehat{J}v\|_2^2 = v^\top \widehat{J}^\top \widehat{J}v = v^\top Cv.$$

Writing

$$v = \sum_{i=1}^{d_a} q_i v_i, \qquad \sum_{i=1}^{d_a} q_i^2 = 1,$$

gives

$$v^\top Cv = \sum_{i=1}^{d_a} \lambda_i q_i^2 \leq \lambda_1.$$

The bound is attained by $v = v_1$. If v is additionally constrained to be orthogonal to $v_1, \ldots, v_{k-1}$, then $q_1 = \cdots = q_{k-1} = 0$, and the same argument gives

$$v^\top Cv \leq \lambda_k,$$

with equality at $v = v_k$.

For each eigenmode,

$$\|\widehat{J}v_k\|_2^2 = v_k^\top Cv_k = \lambda_k.$$

Finally, for $j \neq k$,

$$(\widehat{J}v_j)^\top (\widehat{J}v_k) = v_j^\top Cv_k = \lambda_k v_j^\top v_k = 0.$$

□

Since

$$\operatorname{rank}(\widehat{J}^\top \widehat{J}) = \operatorname{rank}(\widehat{J}) \leq \min(d_z, d_a),$$

there are at most $\min(d_z, d_a)$ modes with nonzero predicted response. Repeated eigenvalues may yield non-unique individual modes within the corresponding eigenspace, without affecting the above properties.

B.2 Policy Improvement Does Not Guarantee Beneficial ARE Rediscovery

Consider DARE with $\alpha = 1$, $K = d_a = 2$, $\mathcal{E}_b = 1$, and exact local response maps in a fixed, unweighted representation. A response map J_i discovered under policy π_i yields the stationary perturbation covariance

$$\mathcal{G}_i = \frac{J_i^\top J_i}{\|J_i\|_F^2}.$$

Let $F_0(\pi)$ denote expected reward without added perturbations, and let $F(\pi, \mathcal{G})$ denote expected reward with $b \sim \mathcal{N}(0, \mathcal{G})$ added before action clipping.

Proposition 2 (Rediscovery need not improve reward)**.** *There exist a task and policies π_0, π_1 such that*

$$F_0(\pi_1) > F_0(\pi_0), \qquad F(\pi_1, \mathcal{G}_1) < F(\pi_1, \mathcal{G}_0).$$

Thus, improving the actor does not guarantee that rediscovering ARE improves collection reward.

Proof. Consider a one-step task with executed action and next state

$$a = \mathrm{clip}(\pi(s) + b, -1, 1), \qquad s^+ = (x, y)^\top = (a_1, a_2 + a_2^2)^\top.$$

Here, a_1, a_2 are the executed action components: the first directly sets x, while the second determines y nonlinearly. Use $z(\tau) = s^+$, so both state coordinates enter response discovery, but define the task reward using only x:

$$r(s^+) = \mathbf{1}\{x \geq 1/4\} + 2\mathbf{1}\{x \geq 3/4\}.$$

The indicator equals one when its condition holds and zero otherwise. Reaching the first threshold earns 1, and reaching the second earns an additional 2, giving total rewards of 0, 1, or 3.

Take the constant policies

$$\pi_0(s) = (0, 0)^\top, \qquad \pi_1(s) = (1/2, 1/2)^\top.$$

Without perturbations, π_0 reaches neither threshold, whereas π_1 reaches the first. Hence $F_0(\pi_0) = 0 < F_0(\pi_1) = 1$.

Next, compute the ARE discovered under each policy. Both actor outputs lie inside the action bounds, so clipping is inactive for sufficiently small probe offsets around zero. Differentiating a_1 and $a_2 + a_2^2$ at the respective actor outputs gives

$$J_0 = \mathrm{diag}(1, 1), \qquad J_1 = \mathrm{diag}(1, 2).$$

The corresponding old and fresh covariances are therefore

$$\mathcal{G}_0 = \mathrm{diag}(1/2, 1/2), \qquad \mathcal{G}_1 = \mathrm{diag}(1/5, 4/5).$$

To isolate the effect of rediscovery, now hold π_1 fixed and change only the perturbation covariance. Writing $b_1 \sim \mathcal{N}(0, \sigma^2)$ gives $x = \mathrm{clip}(1/2 + b_1, -1, 1)$. Both reward thresholds are inside the action bounds, so

$$x \geq 1/4 \iff b_1 \geq -1/4, \qquad x \geq 3/4 \iff b_1 \geq 1/4.$$

Using Gaussian symmetry, the expected reward is

$$\begin{aligned} F(\pi_1, \mathcal{G}) &= \Pr(b_1 \geq -1/4) + 2\Pr(b_1 \geq 1/4) \\ &= 2 - \Phi\left(\frac{1}{4\sigma}\right), \end{aligned}$$

where Φ is the standard normal cumulative distribution function. The old and fresh first-coordinate variances are $1/2$ and $1/5$, respectively. Since Φ is strictly increasing,

$$F(\pi_1, \mathcal{G}_1) = 2 - \Phi\left(\frac{\sqrt{5}}{4}\right) < 2 - \Phi\left(\frac{\sqrt{2}}{4}\right) = F(\pi_1, \mathcal{G}_0).$$

This establishes the claim. □

Rediscovery shifts perturbation energy toward the more responsive second coordinate, reducing exploration along the task-relevant first coordinate. The result concerns collection reward with the updated actor held fixed, not the benefits of subsequent learning.

Table 13: Allegro DAB hyperparameter and representation sweep.

Protocol / arm	P	H_{OU}	$\mathcal{E}_b$	L_0	L_1	L_2	L_3	L_4
Alpha sweep ($\alpha = 0.1$)	16	8	1.00	4789.3 ± 100.6	2791.7 ± 170.6	949.3 ± 34.2	241.3 ± 18.9	64.3 ± 4.0
Alpha sweep ($\alpha = 0.5$)	16	8	1.00	4831.3 ± 48.8	2896.3 ± 132.7	818.0 ± 220.3	174.3 ± 50.3	52.0 ± 11.1
Alpha sweep ($\alpha = 1.0$)	16	8	1.00	4832.7 ± 61.6	2847.7 ± 239.5	864.0 ± 282.1	188.7 ± 84.1	54.7 ± 23.4
Energy sweep	16	8	.50	4742.3 ± 162.8	2506.7 ± 240.7	743.3 ± 114.4	173.7 ± 34.0	45.3 ± 12.7
Energy sweep	16	8	1.00	4722.7 ± 153.2	2523.0 ± 277.8	813.7 ± 276.3	190.3 ± 86.6	49.7 ± 12.6
Energy sweep	16	8	1.50	4808.3 ± 61.0	2798.7 ± 267.7	806.0 ± 141.0	194.3 ± 45.1	51.3 ± 12.1
Energy sweep	16	8	2.00	4761.3 ± 56.0	2842.7 ± 106.3	916.7 ± 162.2	230.7 ± 59.7	65.3 ± 15.7
H_{OU} sweep	16	16	1.00	4831.0 ± 21.7	2675.7 ± 73.2	702.7 ± 38.3	163.7 ± 6.1	51.7 ± 3.1
Probe sweep ($P = 16$)	16	8	1.00	4838.0 ± 38.1	2780.0 ± 134.6	825.0 ± 136.0	190.3 ± 22.5	49.7 ± 8.5
Probe sweep ($P = 32$)	32	8	1.00	4836.7 ± 46.0	2689.0 ± 313.8	832.3 ± 209.3	207.3 ± 64.0	49.0 ± 7.5
Probe sweep ($P = 48$, 2/3 seeds)	48	8	1.00	4862.5 ± 57.3	2795.5 ± 398.1	687.0 ± 108.9	149.5 ± 3.5	47.0 ± 0.0

The legacy reference is $P = 32$, $H_{\mathrm{OU}} = 8$, $\mathcal{E}_b = 1.0$; legacy Δ values use that row. The current P16/U128 rows are a different high-update protocol and their Δ values use the Uniform row only as a descriptive reference. They must not be interpreted as a single factorial fit.

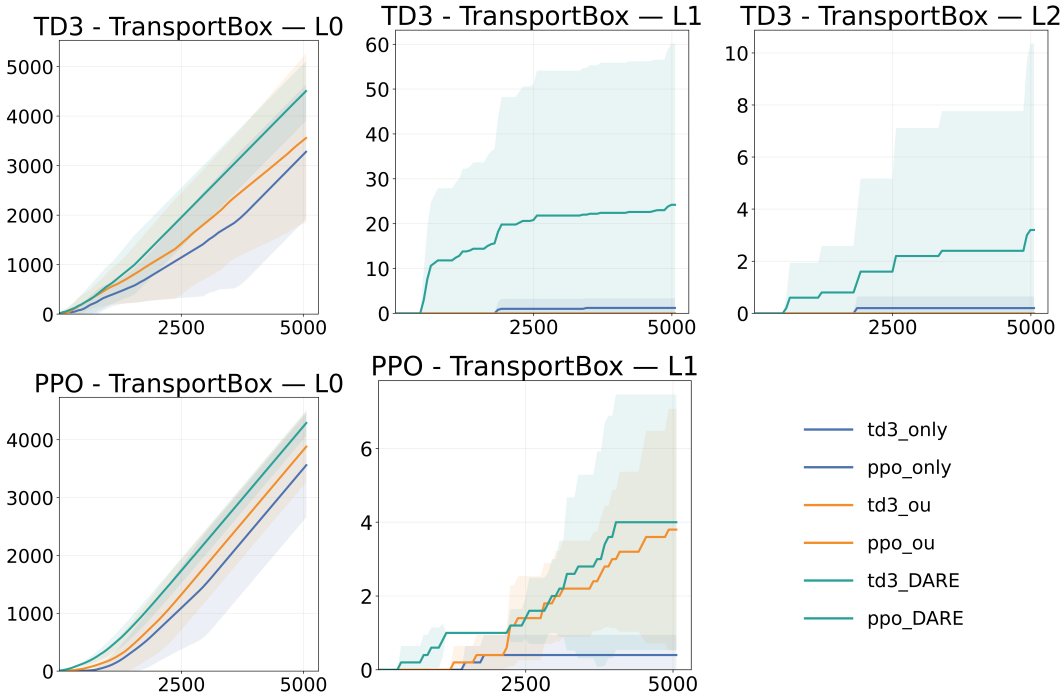
C Detailed Experiment Data

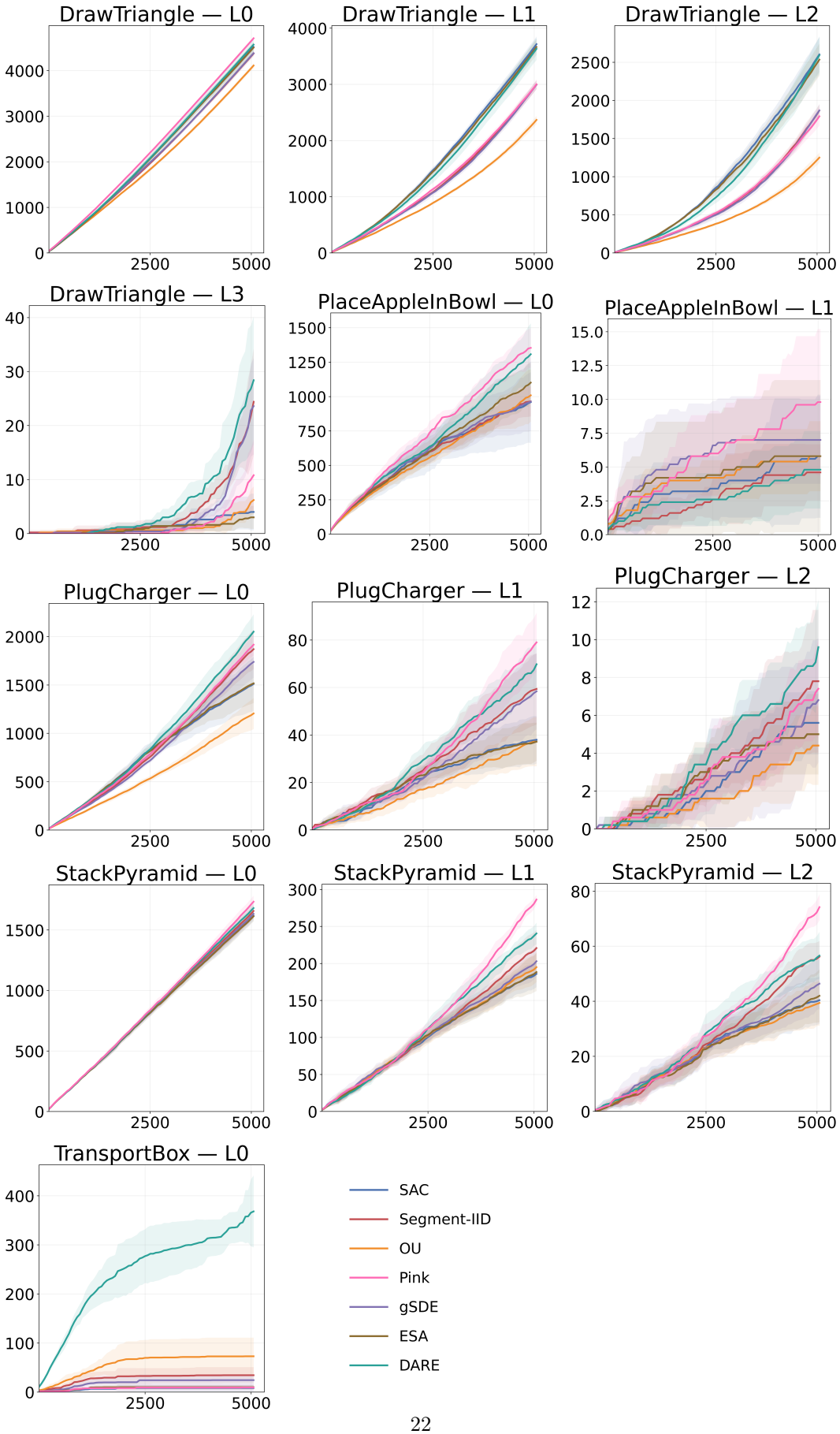
C.1 Hyperparameter Sensitivity

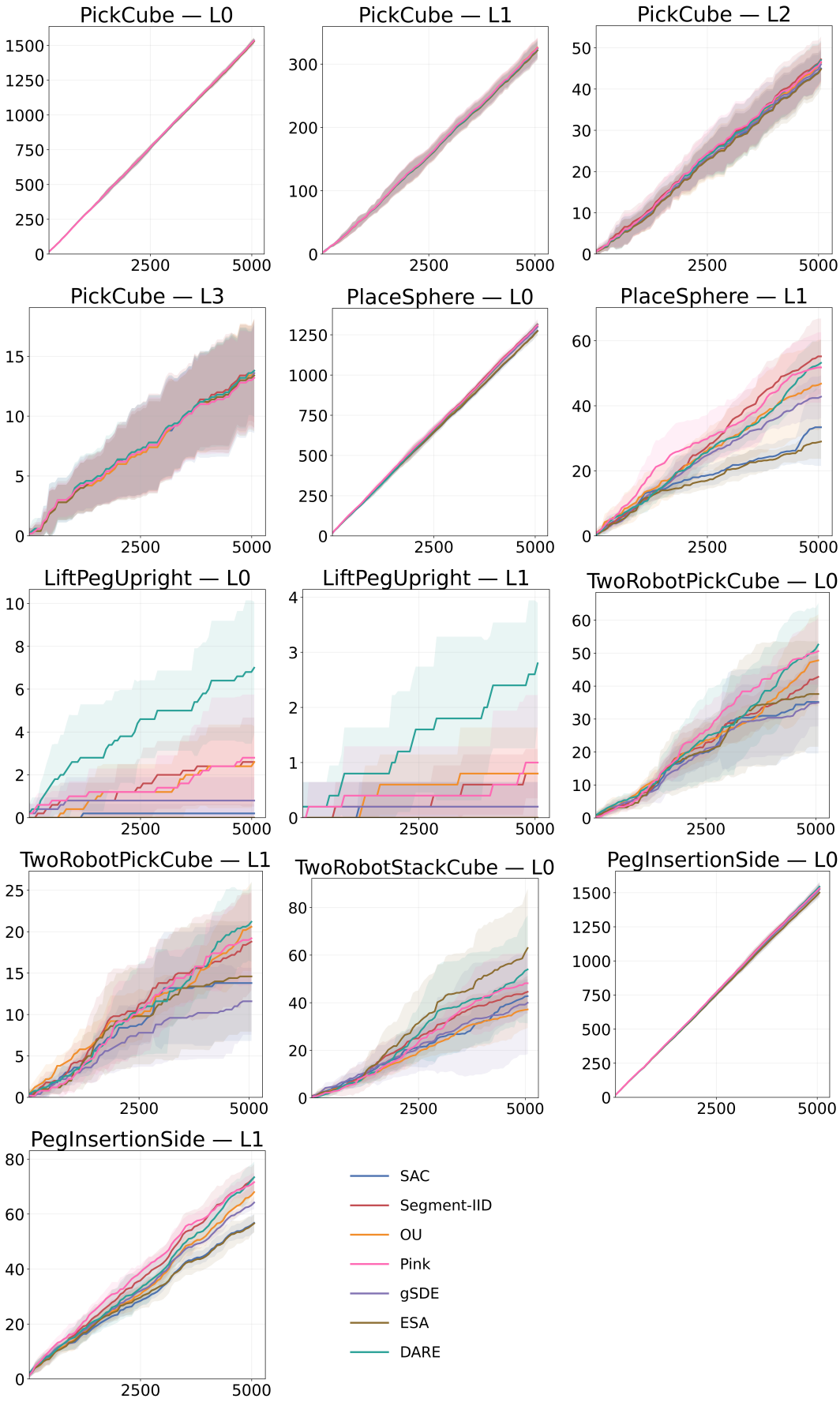
Table 13 varies the OU correlation horizon H_{OU} and exploration energy $\mathcal{E}_b$ on Allegro, changing one parameter at a time from the sweep reference $(H_{\mathrm{OU}}, \mathcal{E}_b) = (8, 1.0)$.

Shortening H_{OU} from 8 to 4 reduces the mean L_3/L_4 counts by approximately 15.7%/18.8%, whereas $H_{\mathrm{OU}} = 16$ yields similar means to the reference. At $H_{\mathrm{OU}} = 8$, $\mathcal{E}_b = 1.50$ gives the highest mean L_3 and L_4 counts, but L_4 does not increase monotonically with energy. The coordinate clipping rate rises from 9.93% at the reference to 15.00% at $\mathcal{E}_b = 1.50$.

C.2 Full Curves




DrawTriangle — L0
DrawTriangle — L1
DrawTriangle — L2
DrawTriangle — L3
PlaceAppleInBowl — L0
PlaceAppleInBowl — L1
PlugCharger — L0
PlugCharger — L1
PlugCharger — L2
StackPyramid — L0
StackPyramid — L1
StackPyramid — L2
TransportBox — L0
SAC
Segment-IID
OU
Pink
gSDE
ESA
DARE


PickCube — L0
PickCube — L1
PickCube — L2
PickCube — L3
PlaceSphere — L0
PlaceSphere — L1
LiftPegUpright — L0
LiftPegUpright — L1
TwoRobotPickCube — L0
TwoRobotPickCube — L1
TwoRobotStackCube — L0
PegInsertionSide — L0
PegInsertionSide — L1
SAC
Segment-IID
OU
Pink
gSDE
ESA
DARE

C.3 Main Result Data

Table 14: Cumulative milestone reaches under SAC (mean ± sample s.d., 5 matched seeds), with bold marking best means with non-overlapping ± s.d. intervals; all additional exploration uses $\mathcal{E}_b = 1.0$. L_4 is omitted as all entries are zero.

Task	Method	L_0	L_1	L_2	L_3
draw_triangle	esa	4504.40 ± 113.36	3665.40 ± 139.16	2535.60 ± 210.43	3.00 ± 2.12
draw_triangle	gsde	4381.80 ± 46.22	2993.40 ± 79.77	1869.40 ± 83.17	23.60 ± 8.91
draw_triangle	ou	4111.20 ± 49.89	2368.40 ± 68.77	1250.60 ± 63.10	6.20 ± 2.49
draw_triangle	pink	4709.00 ± 38.52	3001.60 ± 85.48	1792.00 ± 119.41	10.80 ± 6.57
draw_triangle	DARE	4569.80 ± 126.91	3632.60 ± 196.27	2592.80 ± 245.88	28.40 ± 11.80
draw_triangle	sac_only	4518.80 ± 108.12	3716.00 ± 136.05	2604.00 ± 229.74	4.00 ± 3.39
draw_triangle	segment_iid	4373.60 ± 61.45	2996.20 ± 74.66	1868.60 ± 81.89	24.40 ± 9.91
place_apple_in_bowl	esa	1101.60 ± 142.26	5.80 ± 5.63	0.00 ± 0.00	0.00 ± 0.00
place_apple_in_bowl	gsde	963.60 ± 213.45	7.00 ± 3.24	0.00 ± 0.00	0.00 ± 0.00
place_apple_in_bowl	ou	1009.20 ± 190.08	5.80 ± 2.59	0.00 ± 0.00	0.00 ± 0.00
place_apple_in_bowl	pink	1354.00 ± 144.64	9.80 ± 5.40	0.00 ± 0.00	0.00 ± 0.00
place_apple_in_bowl	DARE	1308.00 ± 223.52	4.80 ± 2.86	0.00 ± 0.00	0.00 ± 0.00
place_apple_in_bowl	sac_only	959.20 ± 295.84	5.80 ± 4.55	0.00 ± 0.00	0.00 ± 0.00
place_apple_in_bowl	segment_iid	965.40 ± 108.81	4.60 ± 2.07	0.00 ± 0.00	0.00 ± 0.00
plug_charger	esa	1517.00 ± 201.65	37.20 ± 7.98	5.00 ± 2.12	0.00 ± 0.00
plug_charger	gsde	1738.40 ± 162.25	58.40 ± 15.95	6.80 ± 3.27	0.00 ± 0.00
plug_charger	ou	1205.60 ± 160.09	37.20 ± 10.33	4.40 ± 2.07	0.00 ± 0.00
plug_charger	pink	1916.40 ± 111.27	79.00 ± 12.43	7.40 ± 2.61	0.00 ± 0.00
plug_charger	DARE	2032.60 ± 234.43	60.40 ± 17.63	8.40 ± 4.34	0.00 ± 0.00
plug_charger	sac_only	1511.00 ± 281.36	38.00 ± 9.97	5.60 ± 1.67	0.00 ± 0.00
plug_charger	segment_iid	1869.80 ± 203.00	59.40 ± 14.98	7.80 ± 3.77	0.00 ± 0.00
stack_pyramid	esa	1612.40 ± 41.66	188.40 ± 25.55	42.00 ± 9.62	0.00 ± 0.00
stack_pyramid	gsde	1633.20 ± 40.64	203.00 ± 20.99	46.40 ± 9.18	0.00 ± 0.00
stack_pyramid	ou	1633.20 ± 33.42	195.00 ± 19.84	39.40 ± 8.17	0.00 ± 0.00
stack_pyramid	pink	1733.20 ± 49.33	286.40 ± 6.11	74.20 ± 4.66	0.00 ± 0.00
stack_pyramid	DARE	1680.60 ± 39.71	240.60 ± 14.45	56.60 ± 8.59	0.00 ± 0.00
stack_pyramid	sac_only	1617.60 ± 40.35	186.00 ± 17.89	40.40 ± 8.62	0.00 ± 0.00
stack_pyramid	segment_iid	1656.80 ± 48.26	220.80 ± 24.51	56.20 ± 5.67	0.00 ± 0.00
transport_box	esa	10.00 ± 4.36	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
transport_box	gsde	24.40 ± 12.18	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
transport_box	ou	73.00 ± 38.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
transport_box	pink	9.40 ± 5.68	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
transport_box	DARE	368.20 ± 72.81	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
transport_box	sac_only	8.40 ± 6.84	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
transport_box	segment_iid	34.40 ± 16.59	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
lift_peg_upright	esa	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
lift_peg_upright	gsde	0.80 ± 1.10	0.20 ± 0.45	0.00 ± 0.00	0.00 ± 0.00
lift_peg_upright	ou	2.60 ± 1.67	0.80 ± 0.84	0.00 ± 0.00	0.00 ± 0.00
lift_peg_upright	pink	2.80 ± 2.95	1.00 ± 1.22	0.20 ± 0.45	0.00 ± 0.00
lift_peg_upright	DARE	7.00 ± 3.08	2.80 ± 1.10	0.80 ± 0.84	0.40 ± 0.55
lift_peg_upright	sac_only	0.20 ± 0.45	0.20 ± 0.45	0.00 ± 0.00	0.00 ± 0.00
lift_peg_upright	segment_iid	2.60 ± 2.07	0.80 ± 0.45	0.20 ± 0.45	0.00 ± 0.00
peg_insertion_side	esa	1500.80 ± 30.65	56.60 ± 3.36	0.00 ± 0.00	0.00 ± 0.00
peg_insertion_side	gsde	1524.00 ± 44.55	64.20 ± 2.39	0.00 ± 0.00	0.00 ± 0.00
peg_insertion_side	ou	1531.00 ± 32.46	68.00 ± 3.00	0.00 ± 0.00	0.00 ± 0.00
peg_insertion_side	pink	1532.60 ± 38.04	71.60 ± 3.44	0.00 ± 0.00	0.00 ± 0.00
peg_insertion_side	DARE	1544.60 ± 34.98	73.40 ± 5.68	0.00 ± 0.00	0.00 ± 0.00
peg_insertion_side	sac_only	1500.60 ± 32.54	56.80 ± 3.11	0.00 ± 0.00	0.00 ± 0.00

Continued on next page

Table 14 – continued from previous page

Task	Method	L_0	L_1	L_2	L_3
peg_insertion_side	segment_iid	1543.40 ± 33.67	73.40 ± 4.83	0.00 ± 0.00	0.00 ± 0.00
pick_cube	esa	1529.00 ± 18.93	321.80 ± 14.96	44.80 ± 4.76	13.40 ± 4.67
pick_cube	gsde	1535.20 ± 19.40	324.80 ± 14.81	46.00 ± 4.06	13.60 ± 4.56
pick_cube	ou	1537.20 ± 21.06	326.20 ± 14.39	46.40 ± 5.03	13.60 ± 4.56
pick_cube	pink	1535.00 ± 17.99	325.60 ± 16.67	46.60 ± 4.77	13.20 ± 4.32
pick_cube	DARE	1537.00 ± 18.84	324.60 ± 12.82	47.00 ± 4.74	13.80 ± 3.83
pick_cube	sac_only	1529.20 ± 17.91	321.60 ± 13.79	45.00 ± 4.06	13.20 ± 4.60
pick_cube	segment_iid	1537.00 ± 18.77	325.00 ± 16.51	47.20 ± 5.50	13.80 ± 4.32
place_sphere	esa	1272.60 ± 23.86	29.00 ± 5.29	0.00 ± 0.00	0.00 ± 0.00
place_sphere	gsde	1299.00 ± 15.56	42.80 ± 6.53	0.20 ± 0.45	0.00 ± 0.00
place_sphere	ou	1302.00 ± 30.22	46.80 ± 13.65	0.20 ± 0.45	0.00 ± 0.00
place_sphere	pink	1310.40 ± 36.75	51.80 ± 10.80	0.00 ± 0.00	0.00 ± 0.00
place_sphere	DARE	1313.20 ± 29.42	53.20 ± 7.16	0.20 ± 0.45	0.20 ± 0.45
place_sphere	sac_only	1275.60 ± 30.37	33.40 ± 11.84	0.00 ± 0.00	0.00 ± 0.00
place_sphere	segment_iid	1315.60 ± 13.16	55.20 ± 11.65	0.00 ± 0.00	0.00 ± 0.00
two_robot_pick_cube	esa	37.60 ± 15.98	14.60 ± 6.66	0.00 ± 0.00	0.00 ± 0.00
two_robot_pick_cube	gsde	35.00 ± 7.18	11.60 ± 4.04	0.20 ± 0.45	0.00 ± 0.00
two_robot_pick_cube	ou	47.80 ± 13.92	20.60 ± 5.41	0.20 ± 0.45	0.00 ± 0.00
two_robot_pick_cube	pink	50.60 ± 10.19	19.20 ± 5.85	0.00 ± 0.00	0.00 ± 0.00
two_robot_pick_cube	DARE	52.60 ± 12.58	21.20 ± 4.82	0.20 ± 0.45	0.00 ± 0.00
two_robot_pick_cube	sac_only	35.20 ± 15.61	13.80 ± 6.98	0.00 ± 0.00	0.00 ± 0.00
two_robot_pick_cube	segment_iid	42.80 ± 6.57	18.80 ± 5.85	0.00 ± 0.00	0.00 ± 0.00
two_robot_stack_cube	esa	63.00 ± 24.87	0.80 ± 1.30	0.60 ± 0.89	0.00 ± 0.00
two_robot_stack_cube	gsde	40.00 ± 21.54	0.20 ± 0.45	0.20 ± 0.45	0.00 ± 0.00
two_robot_stack_cube	ou	37.20 ± 7.73	0.60 ± 0.55	0.40 ± 0.55	0.00 ± 0.00
two_robot_stack_cube	pink	48.20 ± 12.83	0.40 ± 0.55	0.20 ± 0.45	0.00 ± 0.00
two_robot_stack_cube	DARE	54.00 ± 22.64	0.60 ± 0.89	0.40 ± 0.55	0.00 ± 0.00
two_robot_stack_cube	sac_only	42.80 ± 7.16	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
two_robot_stack_cube	segment_iid	44.60 ± 6.69	0.60 ± 0.55	0.60 ± 0.55	0.00 ± 0.00

C.4 Per-seed Data

Table 15: Per-seed cross-algorithm results on `UnitreeG1TransportBox-v1`. Each entry reports $L_0/L_1/L_2$. L_3 and L_4 are omitted because they are zero for all methods and seeds.

Learner	Method	Seed				
		1	2	3	4	5
PPO	DARE	4163/6/0	4184/9/1	4057/1/0	4537/1/0	4508/3/0
	Base	3867/1/0	3357/1/0	2105/0/0	4082/0/0	4391/0/0
	OU	4095/4/0	3788/3/0	2970/9/0	3892/3/0	4661/0/0
		0	1	2	3	4
TD3	DARE	4735/1/0	4935/82/16	4362/0/0	4976/37/0	3529/1/0
	Base	4098/0/0	4312/1/0	4394/5/1	1892/0/0	1706/0/0
	OU	942/0/0	4869/0/0	4601/0/0	4693/0/0	2689/0/0