EGGS: ENTROPY-GUIDED JOINT GOAL SELECTION FOR LONG-HORIZON UNSUPERVISED GOAL-CONDITIONED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider unsupervised goal-conditioned reinforcement learning, where the agent aims to learn a policy that can reach diverse goals without task-specific rewards. To this end, existing methods select goals using visitation novelty or predicted exploration value to guide the agent toward new states, but when several goals are used in the same collection batch, they may be close to one another, leading to overlapping exploration across rollouts. We introduce **EGGS** (*Entropy-Guided Joint Goal Selection*), which jointly selects goals to reduce redundancy across exploration rollouts. EGGS combines a visitation novelty score motivated by visitation entropy with a mutual-information term that measures distinguishability among the selected goals in a learned temporal representation. Each selected goal guides a Go-Explore rollout, which first moves toward the goal and then continues exploration from the resulting state. Compared with selecting goals using visitation novelty alone, EGGS selects more distinguishable goals and collects trajectories with less overlap while retaining high novelty scores. In controlled 2-D environments, EGGS achieves the highest mean coverage AUC in all six tasks and the highest mean final normalized visitation entropy in five of six, demonstrating rapid coverage and broadly distributed visitation. Across AntMaze and locomotion benchmarks, EGGS attains the highest mean coverage AUC in four of six environments, and its pretrained behaviors support zero-shot goal reaching and sequential multi-goal control.

1 INTRODUCTION

Unsupervised goal-conditioned reinforcement learning (GCRL) aims to learn to reach diverse goals without task-specific rewards. A central challenge is to collect experience that broadens state coverage while supporting reusable goal-conditioned control. Expanding coverage is particularly challenging in long-horizon environments, where reaching under-covered regions can require long sequences of actions through frequently visited states. Go-Explore-style collection addresses this challenge by using a goal-conditioned policy to move toward an exploratory goal and then continuing exploration from the resulting state (Ecoffet et al., 2021; Li & Chen, 2020; Bae et al., 2024). Existing approaches select exploratory goals using criteria such as novelty, temporal distance, reachability, and predicted exploration potential (Pitis et al., 2020; Li & Chen, 2020; Bae et al., 2024; Duan et al., 2024; Hu et al., 2023). However, when selecting goals for multiple rollouts, we must also consider how these goals complement one another. Individually promising goals can direct multiple rollouts toward similar regions, producing redundant experience. This motivates jointly selecting exploratory goals to favor under-covered regions while reducing redundancy across rollouts.

We introduce **EGGS** (*Entropy-Guided Joint Goal Selection*), which jointly selects exploratory goals from the replay buffer using an under-coverage score and distinguishability among the selected goals. The under-coverage score favors goals in less-visited regions, while the distinguishability term discourages selecting similar goals in a temporal representation learned using TLDR's representation objective (Bae et al., 2024). The agent collects one fixed-horizon Go-Explore rollout for each selected goal under the same policy and representation parameters, adds the resulting experience to the replay buffer, and then updates the learner.

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.

We use the first variation of visitation entropy to motivate the under-coverage score. An idealized finite-cell model characterizes the trade-off between favoring under-covered cells and spreading goal selections across distinct cells. Compared with selecting the top-K goals ranked by under-coverage score alone, EGGS selects more distinguishable goals and collects trajectories with less overlap while retaining high under-coverage scores. We evaluate EGGS in controlled 2-D environments, AntMaze, and locomotion tasks, where EGGS achieves the highest mean coverage AUC among the compared methods in all six controlled 2-D environments and four of six AntMaze and locomotion environments. Downstream evaluations further demonstrate that EGGS's learned behaviors remain useful for zero-shot goal reaching and sequential multi-goal control.

Our contributions are threefold. First, we introduce EGGS, which jointly selects exploratory goals by combining under-coverage with distinguishability among the selected goals. Second, we motivate the under-coverage score through visitation entropy and use an idealized finite-cell model to analyze the trade-off between selecting under-covered cells and spreading selections across distinct cells, together with a limiting connection to the Gaussian distinguishability criterion. Third, we empirically demonstrate reduced goal and trajectory redundancy, evaluate EGGS across controlled 2-D, AntMaze, and locomotion environments, and test the reuse of its learned behaviors for downstream goal-conditioned control.

2 Related Work

2.1 Reward-Free Exploration and Unsupervised Skill Discovery

Reward-free RL has approached exploration through both state-space coverage and reusable behavioral diversity. State-coverage methods use entropy or density to favor broad visitation: MEPOL optimizes state-distribution entropy, APT and RE3 derive intrinsic rewards from nonparametric entropy estimates in representation space, RAMP relates cumulative entropy improvement to occupancy divergences, and ExDM uses a learned density model to guide exploration toward under-visited regions (Mutti et al., 2021; Liu & Abbeel, 2021; Seo et al., 2021; Tolguenec et al., 2024; Ying et al., 2026). A complementary line of work learns diverse reusable skills through information-theoretic, dynamics-aware, contrastive, or geometry-constrained objectives, including DIAYN, DADS, CIC, BeCL, LSD, and CeSD (Eysenbach et al., 2019; Sharma et al., 2020; Laskin et al., 2022; Yang et al., 2023; Park et al., 2022; Bai et al., 2024). EGGS adopts this visitation-based view to identify under-covered exploratory goals and focuses on the set-level problem of composing several such goals for the same collection batch.

2.2 Goal Selection in Goal-Conditioned RL

Goal-conditioned RL has studied both how goals are generated or distributed and how candidate goals are prioritized for learning and exploration. HER established achieved-goal relabeling, while RIG, GoalGAN, and Skew-Fit develop autonomous goal generation through learned representations or adaptive goal distributions (Andrychowicz et al., 2017; Nair et al., 2018; Florensa et al., 2018; Pong et al., 2020). MEGA and OMEGA further adapt behavioral goals according to the achieved-goal distribution or a desired goal distribution (Pitis et al., 2020). Other approaches enrich exploratory-goal selection with information beyond visitation frequency: PEG predicts exploration potential beyond a candidate goal, while CE^2 incorporates reachability-aware latent structure into frontier-goal selection (Hu et al., 2023; Duan et al., 2024). Explicit diversification has also appeared in related settings: SVGG maintains an adaptive training-goal curriculum through interacting Stein variational particles, and diversity-based HER methods use determinantal point processes to select diverse replay trajectories and hindsight goals (Castanet et al., 2023; Dai et al., 2021). In EGGS, the object of selection is the set of exploratory goals that will each drive collection under the same learner snapshot; the selector combines under-coverage with distinguishability among those selected goals.

2.3 Go-Explore and Temporal Goal-Conditioned Control

A separate line of work studies how exploratory goals can be realized over long horizons. Go-Explore returns to archived states before resuming exploration to address detachment and derailment

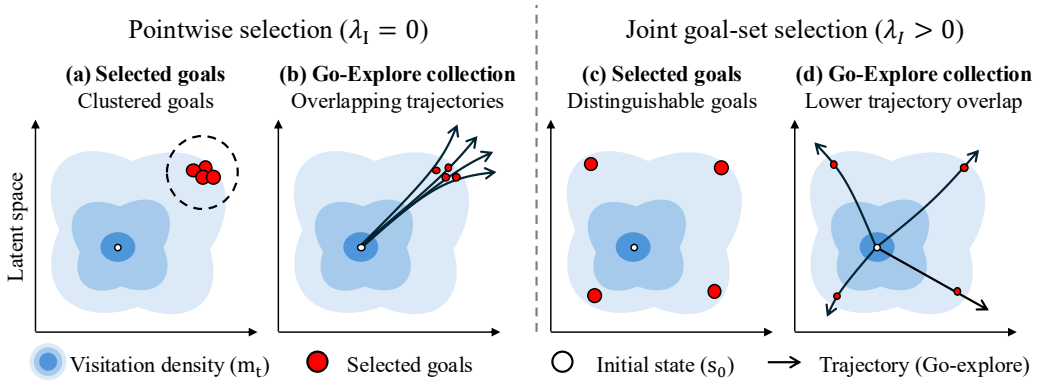


Figure 1: Schematic illustration of pointwise and joint exploratory goal selection in EGGS. (a) With pointwise under-coverage alone ($\lambda_I = 0$), several selected goals can concentrate in the same under-covered region. (b) The corresponding Go-Explore trajectories can exhibit substantial overlap. (c) With $\lambda_I > 0$, the Gaussian information term favors selected goals that remain more distinguishable in the learned temporal representation. (d) The corresponding trajectories can exhibit lower overlap. The candidate pool is omitted and four selected goals are shown for visual clarity. Trajectories are schematic; the Go phase need not terminate exactly at the selected goal.

(Ecoffet et al., 2021). Novelty-Pursuit combines goal-conditioned control toward seldom-visited states with subsequent exploration, while Latent Go-Explore replaces hand-designed cells with learned representations and density estimation (Li & Chen, 2020; Gallouédec & Dellandrea, 2023). Learned state geometries also provide structure for goal selection and control: TLDR uses temporal-distance-aware representations for distant-goal selection, exploration, and goal-conditioned control, while METRA and QRL provide related dynamics-aware geometry through metric-aware abstraction and quasimetric goal-reaching models (Bae et al., 2024; Park et al., 2024; Wang et al., 2023). EGGS adopts TLDR's temporal representation objective within a Go-Explore-style collection procedure, while addressing a different problem: how to jointly select the exploratory goals used within each collection batch.

3 EGGS

EGGS combines three components: an under-coverage score over replay states, a joint distinguishability criterion for selecting multiple exploratory goals, and Go-Explore collection that realizes the selected goals. Figure 1 illustrates how joint goal selection changes the selected goal set and the resulting trajectory overlap.

3.1 Problem Setup and Learning Components

We consider a reward-free MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, \rho_0, \gamma \rangle$, where the agent learns reusable goal-reaching behavior without task-specific rewards. Experience is stored in a replay buffer $\mathcal{D}$, and previously visited states in replay provide candidate exploratory goals for subsequent collection.

At collection cycle t, EGGS maintains a temporal representation $\phi_t : \mathcal{S} \to \mathcal{X} \subseteq \mathbb{R}^d$ learned using TLDR's temporal-distance-aware objective, which ties the latent geometry to environment transitions (Bae et al., 2024). The resulting latent geometry is used to estimate visitation under-coverage, measure distinguishability among selected goals, and define progress toward a selected goal.

EGGS also maintains separate Go and Explore policies. The goal-conditioned Go policy is trained to make temporal-distance progress toward a selected goal, while the Explore policy continues from the state reached by Go using a visitation-based intrinsic reward. After the collected trajectories are added to replay, the temporal representation and both policies are updated from replay before

the next collection cycle. Detailed objectives for the Go and Explore policies are provided in Appendix B.3, with representation-learning and optimization details in Appendix B.4.

3.2 Exploratory Goals and Under-Coverage

The first component of EGGS is an under-coverage score that assigns larger values to candidate goals in less-visited regions. For the analysis, let $m_t(x)$ denote the cumulative visitation density expressed in the current temporal representation, where $x = \phi_t(s)$. In practice, EGGS approximates the corresponding under-coverage signal from retained replay using a kNN density estimator (Appendix B.1).

Under-coverage from visitation entropy. To motivate this score, consider a fixed latent geometry and an infinitesimal change in visitation toward a normalized density ρ_t:

$$m_{t,\alpha} = (1-\alpha)m_t + \alpha\rho_t.$$

Under standard regularity conditions,

$$\frac{d}{d\alpha}H(m_{t,\alpha})\bigg|_{\alpha=0^+} = \mathbb{E}_{x\sim\rho_t}\left[-\log m_t(x)\right] - H(m_t). \tag{1}$$

This motivates

$$u_t(x) := -\log m_t(x)$$

as a state-wise under-coverage utility: lower-density states receive larger first-order preference for additional visitation.

The utility u_t assigns larger scores to less-visited replay states, but evaluates each candidate independently. Selecting the K goals using this pointwise score alone can therefore concentrate multiple goals in similar regions and lead to overlapping collection trajectories. This motivates the joint distinguishability term introduced next.

3.3 Joint Goal Selection in Temporal Geometry

To account for relationships among the selected goals, EGGS augments the pointwise under-coverage utility with a set-level distinguishability term. We measure distinguishability in the learned temporal representation using an auxiliary Gaussian channel, which evaluates whether goal embeddings remain distinguishable after perturbation at scale σ_I.

Let $\mathcal{G}_t$ denote the candidate goals available at collection cycle t, and let

$$\mathcal{B}_t \subseteq \mathcal{G}_t^K$$

denote the feasible selections of K candidate goals. For a batch $\mathbf{g} = (g_1, \ldots, g_K) \in \mathcal{B}_t$ and a uniformly sampled goal index

$$C \sim \mathrm{Unif}\{1, \ldots, K\},$$

define the auxiliary Gaussian channel

$$\tilde{X} \mid C = i \sim \mathcal{N}\big(\phi_t(g_i), \sigma_I^2 I\big). \tag{2}$$

The mutual information $I_{\mathbf{g}}(C; \tilde{X})$ is larger when the selected goal embeddings remain easier to distinguish after perturbation at scale σ_I. Since $I_{\mathbf{g}}(C; \tilde{X}) \le H(C) = \log K$, the criterion approaches its ceiling once the selected components are readily distinguishable at this scale, rather than continuing to reward arbitrarily large pairwise distances. EGGS jointly selects

$$\mathbf{g}_t^\star \in \arg\max_{\mathbf{g}\in\mathcal{B}_t}\left[\frac{1}{K}\sum_{i=1}^{K} u_t(\phi_t(g_i)) + \lambda_I I_{\mathbf{g}}(C; \tilde{X})\right]. \tag{3}$$

The first term evaluates each candidate goal independently, whereas the mutual-information term depends on the selected goals jointly. With $\lambda_I = 0$, Eq. (3) reduces to maximizing the sum of pointwise under-coverage utilities subject to the feasibility constraints. For $\lambda_I > 0$, the value of selecting a candidate also depends on the other goals selected in the same collection cycle.

In practice, EGGS approximately optimizes Eq. (3) over replay candidates using discrete coordinate descent. The density estimator, candidate construction, and optimization procedure are detailed in Appendices B.1 and B.2.

3.4 GO-EXPLORE COLLECTION

Each selected goal is realized through a fixed-horizon Go-Explore rollout: the Go phase uses the goal-conditioned policy to move toward the goal, and the Explore phase continues from the reached state using the exploration policy to collect additional experience. Detailed reward definitions and rollout horizons are provided in Appendix B.3.

At each collection cycle, EGGS first selects all K exploratory goals using Eq. (3). It then collects one Go-Explore rollout for each selected goal and appends the resulting transitions to replay. The temporal representation and the Go and Explore policies are then updated from the augmented replay buffer, after which the next collection cycle begins. Algorithm 1 summarizes this procedure.

3.5 FINITE-CELL ANALYSIS

We use an idealized finite-cell model to isolate the allocation principle behind the EGGS selector: how a finite batch of selections should be distributed across regions with different visitation counts, and how set-level diversity changes that allocation. The model assumes direct allocation to cells and abstracts away the replay candidate pool, goal-reaching dynamics, and learning updates. Consider M cells with cumulative visitation counts $n = (n_1, \dots, n_M)$ and total count $N = \sum_i n_i$. Suppose that K additional selections can be assigned directly across these cells. The finite-budget entropy-allocation problem is

$$\max_{\substack{a \in \mathbb{Z}_{\geq 0}^M \\ \mathbf{1}^\top a = K}} H\left(\frac{n+a}{N+K}\right). \tag{4}$$

The discrete marginal gain of entropy decreases with the current cell count. Consequently, an exact integer optimizer can be obtained by assigning each additional unit sequentially to a currently minimum-count cell. Every optimal allocation also satisfies

$$n_i < n_j \quad \Longrightarrow \quad a_i^\star \geq a_j^\star, \tag{5}$$

which we call *balancing*. The continuous relaxation yields the corresponding water-filling solution $a_i^\star = [\tau - n_i]_+$. Full derivations are provided in Appendix A.1.

Pointwise utility and set-level diversity. The exact entropy-allocation problem above isolates the allocation induced by visitation entropy, whose optimum assigns additional units toward lower-count cells. To study an idealized analogue of the two terms in the EGGS selector, we next combine pointwise under-coverage utility with categorical diversity in the same finite-cell setting. Define the smoothed visitation frequency

$$\tilde{p}_i(n) = \frac{n_i + \eta}{N + M\eta}, \qquad \eta > 0,$$

and let $b_i \in \mathbb{Z}_{\geq 0}$ denote the number of the K selected goals assigned to cell i, with $\sum_i b_i = K$. We consider

$$J_\lambda(b; n) = \frac{1}{K} \sum_{i=1}^{M} b_i \left[-\log \tilde{p}_i(n)\right] + \lambda H(b/K), \qquad \lambda \geq 0. \tag{6}$$

The first term measures the average pointwise under-coverage utility of the selected goals, while the second measures categorical diversity across the selected set.

Every global optimum of Eq. (6) satisfies

$$n_i < n_j \quad \Longrightarrow \quad b_i^\star \geq b_j^\star.$$

To characterize the role of the diversity weight, define

$$L(b; n) := \frac{1}{K} \sum_i b_i [-\log \tilde{p}_i(n)].$$

For $0 \leq \lambda_1 < \lambda_2$, let $b_1^\star$ and $b_2^\star$ be arbitrary global optima at the respective weights. Then

$$H(b_2^\star/K) \geq H(b_1^\star/K), \qquad L(b_2^\star; n) \leq L(b_1^\star; n). \tag{7}$$

At a global optimum of the categorical objective, increasing the diversity weight cannot decrease set entropy or increase average under-coverage utility. Proofs are provided in Appendix A.2.

To connect the categorical analysis to EGGS's continuous Gaussian distinguishability, note that categorical diversity has a simple information-theoretic interpretation. Let $C \sim \mathrm{Unif}\{1, \ldots, K\}$ index a selected goal and let A_{cell} denote its cell. Since A_{cell} is determined by C and $\Pr(A_{\mathrm{cell}} = i) = b_i/K$,

$$I(C; A_{\mathrm{cell}}) = H(A_{\mathrm{cell}}) = H(b/K). \tag{8}$$

This identity concerns the fixed partition used in the analysis. Under an idealized construction in which goals within each cell share a latent center, the same categorical entropy is recovered as the well-separated-cell limit of the Gaussian distinguishability criterion used by EGGS; Appendix A.2 gives the short argument.

Both finite-cell objectives satisfy the same balancing order. Together, these results provide a design-level interpretation of the under-coverage–diversity trade-off underlying the EGGS goal-selection objective. In the fixed finite-cell model, repeated balancing further drives normalized occupancy toward the uniform distribution; the formal statement and proof are provided in Appendix A.3.

4 Experiments and Results

We evaluate EGGS through controlled 2-D exploration, AntMaze and locomotion exploration, component and mechanism analyses, and downstream reuse of pretrained behaviors.

4.1 Experimental Setup

For exploration experiments, we use task-appropriate coverage measures. For AntMaze and controlled 2-D navigation, coverage is the fraction of planar-position cells visited by environment step t. For locomotion, we use continuous positional coverage, $C(t) = x_{\max}(t) - x_{\min}(t)$, where the extrema are accumulated over the interaction history up to step t. We report final coverage $C(T)$ and the time-normalized coverage AUC,

$$\mathrm{AUC}_{\mathrm{cov}} := \frac{1}{T} \int_0^T C(t)\, dt,$$

with coverage AUC as the primary measure of interaction-efficient support expansion.

For the controlled 2-D environments, we additionally measure how cumulative visitation is distributed over the valid state-space grid. Let $n_c(t)$ be the cumulative visit count of valid cell c and

$$p_c(t) = \frac{n_c(t)}{\sum_j n_j(t)}, \qquad \bar{H}_{\mathrm{vis}}(t) = -\frac{\sum_{c=1}^{M} p_c(t) \log p_c(t)}{\log M},$$

where M is the number of valid cells. Coverage measures the extent of the visited support, whereas $\bar{H}_{\mathrm{vis}}$ additionally reflects how cumulative interaction mass is distributed across the state space. We use it as a complementary measure of visitation distribution. For readability, Table 1 reports $100\, \bar{H}_{\mathrm{vis}}(T)$ as a percentage.

Within each environment, all directly compared methods use the same reporting horizon and evaluation protocol. We report AntMaze Large, Medium, and U-Maze over 10M, 5M, and 3M interactions, respectively, and all locomotion environments over 5M interactions. Computationally inexpensive 2-D experiments use 10 seeds; seed counts for the high-dimensional evaluations are reported with the corresponding results. Network architectures, optimization settings, and training hyperparameters are provided in Appendix B.4. Environment configurations and evaluation protocols are provided in Appendix C.

4.2 Controlled 2-D Exploration

We compare EGGS with TLDR (Bae et al., 2024), METRA (Park et al., 2024), ExDM (Ying et al., 2026), RND (Burda et al., 2019), DIAYN (Eysenbach et al., 2019), and CIC (Laskin et al., 2022) in

Table 1: Controlled 2-D exploration over 10 seeds. (a) Time-normalized coverage AUC (%) measures support expansion over the interaction budget. (b) Final normalized cumulative visitation entropy measures the spatial distribution of cumulative visitation. Values are mean ± sample standard deviation. Bold and underlined entries mark the highest and second-highest unrounded means within each environment.

(a) Coverage AUC (%)

Method	Square-A	Square-B	Square-C	Square-D	Square-Tree	Square-Large
TLDR	81.6 ± 2.7	87.0 ± 2.6	65.9 ± 2.3	92.6 ± 0.9	78.4 ± 2.2	37.9 ± 2.1
METRA	75.9 ± 2.8	67.8 ± 4.1	54.4 ± 3.2	83.8 ± 3.0	66.8 ± 2.9	34.3 ± 1.1
RND	84.5 ± 1.9	75.6 ± 3.9	73.0 ± 4.6	86.0 ± 2.0	69.9 ± 3.7	37.6 ± 2.6
DIAYN	74.5 ± 4.4	68.5 ± 7.4	60.0 ± 7.7	84.6 ± 2.9	59.4 ± 3.0	31.2 ± 3.9
CIC	90.7 ± 3.7	87.3 ± 7.9	75.3 ± 4.2	94.2 ± 1.9	80.5 ± 3.9	39.7 ± 2.5
ExDM	$\underline{94.1 \pm 1.8}$	$\underline{91.9 \pm 4.4}$	$\underline{77.3 \pm 7.6}$	$\underline{95.7 \pm 0.9}$	$\underline{81.6 \pm 3.2}$	$\underline{46.5 \pm 2.5}$
EGGS	$\mathbf{95.8 \pm 1.4}$	$\mathbf{97.3 \pm 1.1}$	$\mathbf{93.8 \pm 3.4}$	$\mathbf{96.6 \pm 1.1}$	$\mathbf{94.0 \pm 0.7}$	$\mathbf{54.7 \pm 3.1}$

(b) Final normalized cumulative visitation entropy (%)

Method	Square-A	Square-B	Square-C	Square-D	Square-Tree	Square-Large
TLDR	74.9 ± 2.1	84.6 ± 1.7	79.0 ± 1.6	85.2 ± 1.0	84.0 ± 1.3	69.7 ± 1.6
METRA	66.2 ± 2.0	62.5 ± 2.6	61.7 ± 1.1	72.6 ± 0.8	72.3 ± 1.4	63.6 ± 1.9
RND	71.6 ± 1.6	69.8 ± 1.6	73.8 ± 2.6	73.5 ± 1.8	72.4 ± 3.0	66.0 ± 1.6
DIAYN	74.5 ± 3.0	73.8 ± 3.8	73.3 ± 3.2	79.0 ± 3.0	74.2 ± 3.2	64.5 ± 2.1
CIC	$\underline{93.1 \pm 0.4}$	$\underline{93.1 \pm 0.6}$	$\underline{92.1 \pm 0.5}$	$\mathbf{93.9 \pm 0.6}$	$\underline{92.7 \pm 0.4}$	77.7 ± 1.1
ExDM	92.6 ± 0.5	92.4 ± 0.8	90.3 ± 1.3	$\underline{92.9 \pm 0.5}$	89.4 ± 1.1	$\underline{81.9 \pm 1.3}$
EGGS	$\mathbf{94.4 \pm 1.6}$	$\mathbf{95.9 \pm 0.9}$	$\mathbf{94.3 \pm 1.1}$	90.9 ± 1.1	$\mathbf{94.4 \pm 0.5}$	$\mathbf{82.9 \pm 2.0}$

controlled 2-D navigation environments. For EGGS and the reimplemented baselines, hyperparameters are selected on Square-C using coverage AUC and then fixed across the remaining environments. ExDM uses its released 2-D configuration without environment-specific retuning.

Table 1 reports coverage AUC and final normalized cumulative visitation entropy. EGGS has the highest mean coverage AUC in all six environments and the highest mean final normalized visitation entropy in five of six. Together, these results show that EGGS combines rapid support expansion with broadly distributed cumulative visitation across the controlled state spaces.

Final coverage, support-normalized evenness, and additional metric details are reported in Appendix C.1.

4.3 AntMaze and Locomotion Exploration

We evaluate EGGS against TLDR and METRA on AntMaze Large, Medium, and U-Maze, together with HalfCheetah, Hopper, and Walker2d. EGGS's goal-selection hyperparameters are selected on AntMaze Large and fixed for Medium and U-Maze, and the reported locomotion runs retain the same selector configuration. We use the released implementations of TLDR and METRA. Within each environment, the compared methods use the same environment-interaction budget and rollout horizon and are evaluated using the same pre-specified protocol.

AntMaze navigation. AntMaze combines articulated control with long-horizon navigation, requiring repeated traversal of previously visited regions to expand coverage. Figure 2 shows coverage throughout reward-free pretraining. EGGS has the highest mean coverage AUC on Medium and U-Maze, while TLDR has the highest mean on Large. Thus, relative performance varies across the three AntMaze layouts rather than favoring one method uniformly.

Locomotion. We additionally evaluate HalfCheetah, Hopper, and Walker2d under distinct continuous-control dynamics using the same EGGS selector configuration as in AntMaze. Under this selector configuration, EGGS has the highest mean coverage AUC on Hopper and Walker2d,

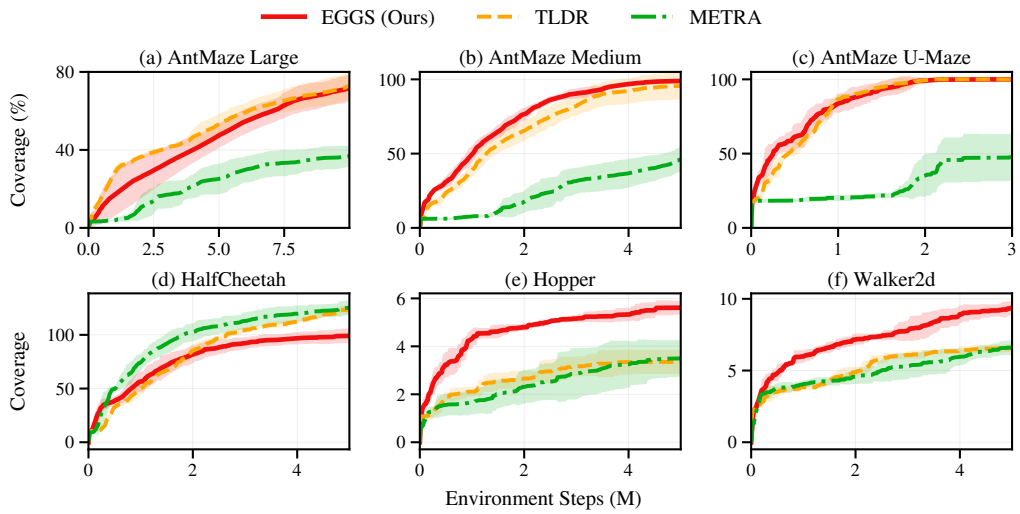


Figure 2: Exploration coverage throughout reward-free pretraining, reported as maze-cell coverage (%) for AntMaze (top) and positional coverage in native units for locomotion (bottom). Curves show mean ± SEM over five independent training seeds.

while METRA has the highest mean on HalfCheetah. Exact coverage AUC and final-coverage values, together with full evaluation details, are reported in Appendix C.2.

4.4 Component Analysis and Mechanism Diagnostics

Component analysis. On AntMaze Large (5M interactions; three matched seeds), mean coverage AUC is 22.4 for Explore-only, 36.3 for single-goal Go-Explore, 38.6 for $K = 8$ without Gaussian distinguishability, and 38.4 with it, yielding no mean AUC gain from the Gaussian term in this setting. In controlled 2-D experiments over 10 paired seeds, Gaussian distinguishability produces negligible or mixed mean changes on Square-A–C and positive mean changes in both coverage AUC and final normalized visitation entropy on Square-D, Square-Tree, and Square-Large, with the largest changes on Square-Large. Full component and sensitivity results are reported in Appendix C.3.

Mechanism diagnostic. We examine how goal-selection criteria trade off under-coverage utility and diversity on matched AntMaze Large snapshots (Table 2). We compare pointwise Top-K, EGGS, and greedy farthest-point selection (FPS) using identical candidate goals, frozen representations, and frozen Go-Explore policies. Goal-set properties are measured before collection; trajectory Jaccard overlap and the number of distinct visited cells are measured after collection, without learning updates. Trajectory overlap is computed over planar cell sets, and Distinct cells counts the union of cells visited by the trajectories in the diagnostic batch, regardless of whether those cells were visited earlier in training.

Compared with pointwise Top-K selection, EGGS retains near-top under-coverage utility (99.09 versus 99.66 percentile) while reducing trajectory Jaccard overlap from 0.231 to 0.134 and increasing distinct visited cells from 111.63 to 130.39. FPS yields still lower overlap and more distinct visited cells, while selecting substantially lower-utility goals. The three selectors therefore occupy different operating points between pointwise under-coverage utility and geometric diversification.

Additional parameter sensitivity results and the complete mechanism-diagnostic protocol and metrics are provided in Appendix C.3 and Appendix C.4.

4.5 Downstream Reuse of Pretrained Behaviors

We evaluate reuse of the frozen pretrained behaviors through zero-shot goal reaching and sequential multi-goal hierarchical RL (HRL) on AntMaze Medium and Large. Table 3 summarizes both

Table 2: Matched-snapshot selector diagnostic on AntMaze Large at 6M interactions. All selectors share candidate pools, a frozen representation, and frozen Go-Explore policies; FPS starts from the highest-utility candidate and greedily maximizes minimum latent distance to the selected set. Goal-set metrics are measured before collection and trajectory metrics after frozen one-batch collection. Values are mean ± sample SD across five training seeds; for $K = 8$, $\max H(b/K) = \log 8 = 2.079$.

	Selected goal set			Collected trajectories	
Selector	Goal-cell $H(b/K)\uparrow$	$d_{\min}/\sigma_I \uparrow$	Mean utility pct. ↑	Jaccard overlap ↓	Distinct cells ↑
Pointwise Top-K	1.925 ± 0.104	0.178 ± 0.091	$\mathbf{99.66 \pm 0.00}$	0.231 ± 0.044	111.63 ± 9.49
EGGS	2.031 ± 0.014	0.732 ± 0.238	99.09 ± 0.16	0.134 ± 0.004	130.39 ± 5.49
FPS	$\mathbf{2.079 \pm 0.000}$	$\mathbf{2.826 \pm 0.154}$	82.30 ± 2.02	$\mathbf{0.111 \pm 0.002}$	$\mathbf{136.75 \pm 7.09}$

Table 3: Downstream reuse of frozen pretrained behaviors on AntMaze Medium and Large. Zero-shot evaluation reports success rate (SR) and normalized geodesic progress P_{geo}; sequential multi-goal HRL reports final goals per episode and Goals AUC. Values are mean ± sample SD over five independently pretrained seeds; higher is better.

		Zero-shot goal reaching		Sequential HRL	
Environment	Method	SR (%)	P_{geo}	Goals / episode	Goals AUC
Medium	TLDR	$\mathbf{83.7 \pm 5.8}$	0.927 ± 0.024	$\mathbf{2.41 \pm 0.14}$	$\mathbf{1.931 \pm 0.049}$
	METRA	20.8 ± 5.8	0.689 ± 0.079	0.89 ± 0.29	0.704 ± 0.221
	EGGS	83.5 ± 5.9	$\mathbf{0.932 \pm 0.025}$	2.20 ± 0.32	1.690 ± 0.250
Large	TLDR	48.1 ± 3.0	0.800 ± 0.027	2.39 ± 0.04	$\mathbf{1.973 \pm 0.033}$
	METRA	14.7 ± 5.1	0.650 ± 0.064	1.66 ± 0.52	1.525 ± 0.469
	EGGS	$\mathbf{53.0 \pm 6.7}$	$\mathbf{0.835 \pm 0.027}$	$\mathbf{2.42 \pm 0.22}$	1.925 ± 0.082

evaluations. In zero-shot goal reaching, TLDR has the higher mean success rate on Medium while EGGS has the higher mean geodesic progress; on Large, EGGS has the highest mean on both metrics. In sequential HRL, TLDR has the highest mean final goals and Goals AUC on Medium, while on Large EGGS has the highest mean final goals and TLDR the highest Goals AUC. Full protocols are provided in Appendix C.5.

5 Conclusion and Limitations

We introduced EGGS for batched long-horizon reward-free exploration, jointly selecting exploratory goals using pointwise under-coverage and distinguishability in a learned temporal representation and realizing each through a Go-Explore rollout. An idealized finite-cell analysis characterizes entropy-optimal allocation toward less-visited regions and the trade-off between pointwise under-coverage and diversity.

Matched AntMaze Large diagnostics show that EGGS substantially reduces goal and trajectory redundancy relative to pointwise selection while retaining high under-coverage utility; greedy farthest-point selection diversifies more aggressively but at a substantially larger utility cost. Separately, controlled 2-D evaluations show that EGGS has the highest mean coverage AUC in all six environments and the highest mean final normalized visitation entropy in five of six, while the AntMaze and locomotion results exhibit environment-dependent relative performance. EGGS's pretrained behaviors also support zero-shot goal reaching and sequential multi-goal downstream control.

Limitations. The finite-cell analysis abstracts away the learned temporal representation, control dynamics, and trajectory-level collection used by EGGS, and therefore characterizes the allocation structure rather than the full learning dynamics. The current temporal geometry uses symmetric Euclidean distance and may not capture directional reachability in partially irreversible environments. Finally, we evaluate interaction efficiency under batched collection; sequential single-goal collection with intermediate updates and end-to-end wall-clock cost are not evaluated.

AI USE STATEMENT

The authors used generative AI tools, including ChatGPT (OpenAI), for assistance with mathematical proofs, feedback on research methodology and experimental design, code development and debugging, interpretation of experimental results, literature search, and manuscript drafting and editing. All AI-assisted outputs were reviewed and independently verified as appropriate; the authors made the final methodological and scientific decisions and take full responsibility for the work.

REPRODUCIBILITY STATEMENT

The main algorithm and theoretical assumptions are specified in the main text, with complete proofs provided in Appendix A. Implementation details, including goal selection, Go-Explore collection, optimization, replay sampling, and hyperparameters, are provided in Appendix B.1 and the following subsections. Experimental protocols and additional evaluation details are provided in Appendix C. The implementation is based in part on the publicly available TLDR codebase. We plan to release the code used for the experiments upon acceptance.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Junik Bae, Kwanyoung Park, and Youngwoon Lee. TLDR: Unsupervised goal-conditioned RL via temporal distance-aware representations. In *Conference on Robot Learning*, 2024.

Chenjia Bai, Rushuai Yang, Qiaosheng Zhang, Kang Xu, Yi Chen, Ting Xiao, and Xuelong Li. Constrained ensemble exploration for unsupervised skill discovery. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 2418–2442. PMLR, 2024.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by Random Network Distillation. In *International Conference on Learning Representations*, 2019.

Victor Campos, Alexander Trott, Caiming Xiong, Richard Socher, Xavier Giro-I-Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1317–1327. PMLR, 2020.

Nicolas Castanet, Olivier Sigaud, and Sylvain Lamprier. Stein variational goal generation for adaptive exploration in multi-goal reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 3714–3731. PMLR, 2023.

Tianhong Dai, Hengyan Liu, Kai Arulkumaran, Guangyu Ren, and Anil Anthony Bharath. *Diversity-Based Trajectory and Goal Selection with Hindsight Experience Replay*, pp. 32–45. Springer International Publishing, 2021. ISBN 9783030893705. doi: 10.1007/978-3-030-89370-5_3.

Yuanlin Duan, Guofeng Cui, and He Zhu. Exploring the edges of latent state clusters for goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 61160–61191. Curran Associates, Inc., 2024. doi: 10.52202/079017-1955.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021. ISSN 1476-4687. doi: 10.1038/s41586-020-03157-9.

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.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1515–1528. PMLR, 2018.

Quentin Gallouédec and Emmanuel Dellandrea. Cell-free latent go-explore. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10571–10586. PMLR, 2023.

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*, pp. 1861–1870. PMLR, 2018.

Edward S. Hu, Richard Chang, Oleh Rybkin, and Dinesh Jayaraman. Planning goals for exploration. In *The Eleventh International Conference on Learning Representations*, 2023.

Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Unsupervised reinforcement learning with contrastive intrinsic control. In *Advances in Neural Information Processing Systems*, 2022.

Ziniu Li and Xiong-Hui Chen. Efficient exploration by novelty-pursuit. In *International Conference on Distributed Artificial Intelligence*, pp. 85–102. Springer, 2020.

Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18459–18473. Curran Associates, Inc., 2021.

Mirco Mutti, Lorenzo Pratissoli, and Marcello Restelli. Task-agnostic exploration via policy gradient of a non-parametric state entropy estimate. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9028–9036, 2021.

Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Seohong Park, Jongwook Choi, Jaekyeom Kim, Honglak Lee, and Gunhee Kim. Lipschitz-constrained unsupervised skill discovery. In *International Conference on Learning Representations*, 2022.

Seohong Park, Oleh Rybkin, and Sergey Levine. METRA: Scalable unsupervised RL with metric-aware abstraction. In *The Twelfth International Conference on Learning Representations*, 2024.

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*, pp. 7750–7761. PMLR, 2020.

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*, pp. 7783–7792. PMLR, 2020.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9443–9454. PMLR, 2021.

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

Paul-Antoine Le Tolguenec, Yann Besse, Florent Teichteil-Koenigsbuch, Dennis G. Wilson, and Emmanuel Rachelson. Exploration by running away from the past, 2024.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36411–36430. PMLR, 2023.

Rushuai Yang, Chenjia Bai, Hongyi Guo, Siyuan Li, Bin Zhao, Zhen Wang, Peng Liu, and Xuelong Li. Behavior contrastive learning for unsupervised skill discovery. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 39183–39204. PMLR, 2023.

Chengyang Ying, Huayu Chen, Xinning Zhou, Zhongkai Hao, Hang Su, and Jun Zhu. Exploratory diffusion model for unsupervised reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

A Theory

A.1 Exact Finite-Budget Entropy Allocation

This section provides the proofs for the finite-cell allocation analysis presented in Sec. 3.5. We first prove the exact integer result for a budget of K unit allocations and then derive its continuous relaxation.

Integer finite-budget allocation. Let

$$n \in \mathbb{Z}_{\geq 0}^{M}, \qquad N := \sum_{i=1}^{M} n_i,$$

denote the current visitation counts over M cells. For a budget of $K \in \mathbb{Z}_{>0}$ additional units, consider

$$\max_{\substack{a \in \mathbb{Z}_{\geq 0}^{M} \\ \mathbf{1}^\top a = K}} H\left(\frac{n+a}{N+K}\right). \tag{9}$$

Since $N + K$ is fixed, defining $f(k) := -k \log k$ with $f(0) := 0$ makes Eq. (9) equivalent to

$$\max_{\substack{a \in \mathbb{Z}_{\geq 0}^{M} \\ \mathbf{1}^\top a = K}} \sum_{i=1}^{M} f(n_i + a_i).$$

Lemma 1 (Integer entropy allocation)**.** *An optimizer of Eq.* (9) *can be obtained by assigning each additional unit sequentially to a cell with minimum current count. Moreover, every optimal integer allocation $a^\star$ satisfies*

$$n_i < n_j \quad \Longrightarrow \quad a_i^\star \geq a_j^\star. \tag{10}$$

Proof. By strict concavity of f, the discrete marginal gain

$$\delta(k) := f(k+1) - f(k)$$

is strictly decreasing in k. Successive units assigned to cell i therefore yield the decreasing sequence

$$\delta(n_i), \quad \delta(n_i + 1), \quad \delta(n_i + 2), \ldots.$$

Any feasible allocation selects a prefix of this sequence for each cell, with K marginal gains selected in total. Because $\delta(k)$ decreases with the current count, the largest currently available marginal gain is attained by a minimum-count cell. Sequentially assigning each unit to such a cell therefore constructs a global optimum.

To prove Eq. (10), suppose that $n_i < n_j$ but an optimal allocation satisfies $a_i^\star < a_j^\star$. Since the quantities are integer-valued,

$$n_i + a_i^\star \leq n_j + a_j^\star - 2.$$

Consider the feasible allocation

$$a' = a^\star + e_i - e_j.$$

Because $\delta(k)$ is strictly decreasing,

$$f(n_i + a_i^\star + 1) - f(n_i + a_i^\star) > f(n_j + a_j^\star) - f(n_j + a_j^\star - 1).$$

Hence

$$\sum_k f(n_k + a'_k) > \sum_k f(n_k + a_k^\star),$$

contradicting the optimality of $a^\star$. Therefore Eq. (10) holds. □

Continuous relaxation. For comparison, relax the allocation to $a \in \mathbb{R}_+^M$ while retaining the same total budget $\mathbf{1}^\top a = K$:

$$\max_{\substack{a \in \mathbb{R}_+^M \\ \mathbf{1}^\top a = K}} H\left(\frac{n+a}{N+K}\right). \tag{11}$$

Let $y_i := n_i + a_i$ and $S := N + K$. Since

$$H\left(\frac{y}{S}\right) = \log S - \frac{1}{S}\sum_{i=1}^M y_i \log y_i,$$

Eq. (11) is equivalent to

$$\min_{\substack{y_i \geq n_i \\ \sum_i y_i = S}} \sum_{i=1}^M y_i \log y_i.$$

Proposition 1 (Water-filling optimum)**.** *The unique optimizer of Eq.* (11) *is*

$$a_i^\star = [\tau - n_i]_+, \qquad i = 1, \ldots, M, \tag{12}$$

where τ is uniquely determined by

$$\sum_{i=1}^M [\tau - n_i]_+ = K.$$

Proof. At any optimum with $K > 0$, every coordinate satisfies $y_i^\star > 0$. If $y_i^\star = 0$, then $n_i = 0$; transferring a sufficiently small amount of mass from a coordinate with positive added mass to coordinate i preserves feasibility and strictly decreases $\sum_j y_j \log y_j$, since the right derivative of $y \log y$ tends to $-\infty$ as $y \to 0^+$.

The feasible set is convex, Slater's condition holds for $K > 0$, and the objective is strictly convex on the positive orthant. Hence the KKT conditions are necessary and sufficient and characterize the unique optimum. With multiplier ν for the equality constraint and multipliers $\alpha_i \geq 0$ for $y_i \geq n_i$, the Lagrangian is

$$\mathcal{L}(y, \nu, \alpha) = \sum_{i=1}^M y_i \log y_i + \nu\left(\sum_{i=1}^M y_i - S\right) - \sum_{i=1}^M \alpha_i (y_i - n_i).$$

Stationarity gives

$$\log y_i + 1 + \nu - \alpha_i = 0.$$

Define $\tau := e^{-1-\nu}$. If $y_i > n_i$, complementary slackness gives $\alpha_i = 0$, and hence $y_i = \tau$. If $y_i = n_i$, then

$$\alpha_i = \log \frac{n_i}{\tau} \geq 0,$$

so $n_i \geq \tau$. Therefore

$$y_i^\star = \max\{n_i, \tau\},$$

which yields Eq. (12).

Finally, define

$$F(\tau) := \sum_{i=1}^M [\tau - n_i]_+.$$

The function F is continuous and is strictly increasing whenever $F(\tau) > 0$, with $F(\tau) \to \infty$ as $\tau \to \infty$. Since $K > 0$, there is a unique τ satisfying $F(\tau) = K$. □

Since $z \mapsto [\tau - z]_+$ is nonincreasing, the continuous optimum also satisfies

$$n_i < n_j \quad \Longrightarrow \quad a_i^\star \geq a_j^\star. \tag{13}$$

Thus, the exact integer problem and its continuous relaxation obey the same balancing order.

A.2 Categorical Allocation Model

Sec. 3.5 introduced a categorical design model that separates pointwise under-coverage utility from set-level diversity over a finite partition. This section proves the balancing property of its global optima, characterizes the effect of the diversity weight, and records a limiting connection to the Gaussian distinguishability criterion used by EGGS.

Let $n = (n_1, \ldots, n_M)$ denote the current visitation counts of M cells, with

$$N := \sum_{i=1}^{M} n_i.$$

For a positive pseudocount $\eta > 0$, define

$$\tilde{p}_i(n) := \frac{n_i + \eta}{N + M\eta}, \qquad \ell_i(n) := -\log \tilde{p}_i(n).$$

For $K \in \mathbb{Z}_{>0}$ selected goals, define

$$\mathcal{B}_K := \left\{ b \in \mathbb{Z}_{\geq 0}^M : \sum_{i=1}^{M} b_i = K \right\}.$$

The categorical objective is

$$J_\lambda(b; n) = \frac{1}{K} \sum_{i=1}^{M} b_i \ell_i(n) + \lambda H(b/K), \qquad b \in \mathcal{B}_K, \quad \lambda \geq 0. \tag{14}$$

Balancing of categorical optima.

Proposition 2 (Balancing of categorical optima)**.** *Every global optimum*

$$b^\star \in \arg\max_{b \in \mathcal{B}_K} J_\lambda(b; n)$$

satisfies

$$n_i < n_j \quad \Longrightarrow \quad b_i^\star \geq b_j^\star. \tag{15}$$

Proof. If $n_i < n_j$, then $\tilde{p}_i(n) < \tilde{p}_j(n)$ and hence $\ell_i(n) > \ell_j(n)$. Suppose a global optimum satisfies $b_i^\star < b_j^\star$, and let b' be the feasible allocation obtained by swapping only the multiplicities of cells i and j. Because b'/K is a coordinate permutation of $b^\star/K$,

$$H(b'/K) = H(b^\star/K).$$

On the other hand,

$$J_\lambda(b'; n) - J_\lambda(b^\star; n) = \frac{(b_j^\star - b_i^\star)(\ell_i(n) - \ell_j(n))}{K} > 0,$$

contradicting the optimality of $b^\star$. Therefore Eq. (15) holds. □

Effect of the diversity weight. Define

$$L(b; n) := \frac{1}{K} \sum_i b_i \ell_i(n).$$

Proposition 3 (Effect of the diversity weight)**.** *Let $0 \leq \lambda_1 < \lambda_2$, and let*

$$b_1^\star \in \arg\max_{b \in \mathcal{B}_K} J_{\lambda_1}(b; n), \qquad b_2^\star \in \arg\max_{b \in \mathcal{B}_K} J_{\lambda_2}(b; n)$$

be arbitrary global optima. Then

$$H(b_2^\star/K) \geq H(b_1^\star/K), \qquad L(b_2^\star; n) \leq L(b_1^\star; n). \tag{16}$$

Proof. Write

$$H_i := H(b_i^\star/K), \qquad L_i := L(b_i^\star; n).$$

Optimality at λ_1 and λ_2 gives

$$L_1 + \lambda_1 H_1 \geq L_2 + \lambda_1 H_2$$

and

$$L_2 + \lambda_2 H_2 \geq L_1 + \lambda_2 H_1.$$

Adding the two inequalities yields

$$(\lambda_2 - \lambda_1)(H_2 - H_1) \geq 0,$$

so

$$H_2 \geq H_1.$$

Substituting this into the first optimality inequality gives

$$L_1 - L_2 \geq \lambda_1(H_2 - H_1) \geq 0,$$

and therefore

$$L_2 \leq L_1.$$

□

Connection to Gaussian distinguishability. The categorical information term above is related to the Gaussian criterion used by EGGS under a simple limiting construction. Let

$$C \sim \mathrm{Unif}\{1, \ldots, K\}$$

index a selected goal, and let $A = q(C)$ denote its cell in the fixed partition, with

$$\Pr(A = r) = \frac{b_r}{K}.$$

Suppose that all selected goals assigned to cell r share a latent center μ_r, so that

$$\tilde{X} \mid A = r \sim \mathcal{N}(\mu_r, \sigma_I^2 I).$$

Under this construction, $\tilde{X}$ is conditionally independent of C given A, and since A is a deterministic function of C,

$$I(C; \tilde{X}) = I(A; \tilde{X}).$$

Now let

$$d_{\min}^{\mathrm{cell}} := \min_{r \neq s} \|\mu_r - \mu_s\|_2.$$

As

$$\frac{d_{\min}^{\mathrm{cell}}}{\sigma_I} \to \infty,$$

the cell identity A can be recovered from $\tilde{X}$ with vanishing classification error. Because the number of cells occupied by the selected goals is finite, this implies

$$H(A \mid \tilde{X}) \to 0.$$

Therefore,

$$I(C; \tilde{X}) = I(A; \tilde{X}) \longrightarrow H(A) = H(b/K). \tag{17}$$

Thus, under this idealized collapsed-within-cell construction, the categorical diversity term is recovered as the well-separated-cell limit of the Gaussian distinguishability criterion.

A.3 Repeated Balancing

For $n \in \mathbb{R}^M_{\geq 0}$, define the range by

$$\Delta(n) := \max_i n_i - \min_i n_i.$$

We say that an allocation $a \in \mathbb{R}^M_{\geq 0}$ is *balancing* with respect to n if

$$n_i < n_j \quad \Longrightarrow \quad a_i \geq a_j.$$

Theorem 1 (Repeated balancing)**.** *Consider a fixed M-cell domain with $n_0 \in \mathbb{R}^M_{\geq 0}$, $N_0 = \mathbf{1}^\top n_0 > 0$, and $K > 0$. Let*

$$n_{\ell+1} = n_\ell + a_\ell,$$

where every $a_\ell \in \mathbb{R}^M_{\geq 0}$ is balancing with respect to n_ℓ and satisfies $\mathbf{1}^\top a_\ell = K$. With

$$p_\ell = \frac{n_\ell}{N_0 + \ell K}, \qquad \Delta_0 = \max_i n_{0,i} - \min_i n_{0,i},$$

we have

$$\left\| p_\ell - \frac{\mathbf{1}}{M} \right\|_\infty \leq \frac{\max\{\Delta_0, K\}}{N_0 + \ell K} \longrightarrow 0, \qquad H(p_\ell) \longrightarrow \log M. \tag{18}$$

Lemma 2 (One-step range bound)**.** *Let $a \in \mathbb{R}^M_{\geq 0}$ be balancing with respect to n and satisfy $\mathbf{1}^\top a = K$. Then*

$$\Delta(n + a) \leq \max\{\Delta(n), K\}. \tag{19}$$

Proof. Let i and j attain the maximum and minimum of $n + a$, respectively. Then

$$\Delta(n + a) = (n_i + a_i) - (n_j + a_j).$$

If $n_i > n_j$, balancing implies $a_i \leq a_j$, and therefore

$$(n_i + a_i) - (n_j + a_j) \leq n_i - n_j \leq \Delta(n).$$

If $n_i < n_j$, balancing implies $a_i \geq a_j$, and hence

$$(n_i + a_i) - (n_j + a_j) \leq a_i - a_j \leq K,$$

where the last inequality follows from $a \geq 0$ and $\mathbf{1}^\top a = K$. Finally, if $n_i = n_j$, then

$$(n_i + a_i) - (n_j + a_j) = a_i - a_j \leq K.$$

Thus Eq. (19) holds. □

Proof of Theorem 1. Let

$$n_{\ell+1} = n_\ell + a_\ell, \qquad a_\ell \in \mathbb{R}^M_{\geq 0}, \qquad \mathbf{1}^\top a_\ell = K,$$

where every a_ℓ is balancing with respect to n_ℓ. Define

$$\Delta_\ell := \Delta(n_\ell).$$

By Lemma 2,

$$\Delta_{\ell+1} \leq \max\{\Delta_\ell, K\}.$$

Hence, by induction,

$$\Delta_\ell \leq \max\{\Delta_0, K\} \qquad \text{for all } \ell \geq 0.$$

Let

$$N_0 := \mathbf{1}^\top n_0 > 0.$$

The cumulative mass after ℓ rounds is

$$N_\ell := \mathbf{1}^\top n_\ell = N_0 + \ell K.$$

Since the mean cell mass N_ℓ / M lies between the minimum and maximum coordinates of n_ℓ, for every i,

$$\left| n_{\ell,i} - \frac{N_\ell}{M} \right| \leq \Delta_\ell.$$

Therefore, with

$$p_\ell := \frac{n_\ell}{N_\ell},$$

we obtain

$$\left\| p_\ell - \frac{\mathbf{1}}{M} \right\|_\infty \le \frac{\Delta_\ell}{N_\ell} \le \frac{\max\{\Delta_0, K\}}{N_0 + \ell K} \longrightarrow 0. \tag{20}$$

Thus

$$p_\ell \longrightarrow \frac{\mathbf{1}}{M}.$$

By continuity of entropy on the finite probability simplex,

$$H(p_\ell) \longrightarrow H\left(\frac{\mathbf{1}}{M}\right) = \log M.$$

□

B Implementation Details

B.1 Temporal Representation and Visitation Density

EGGS adopts TLDR's temporal-distance-aware representation objective (Bae et al., 2024) and uses the mean of the Gaussian encoder, $\phi(s) = \mu_\theta(s) \in \mathbb{R}^4$, as the latent representation. Representation inputs use fixed environment-specific normalization, while the latent output is used directly without normalization.

For the AntMaze and locomotion experiments, replay has capacity 10^6 transitions and overwrites the oldest entries once the buffer is full. Accordingly, the practical density estimator below is defined over the currently retained replay rather than the full interaction history.

For goal selection, EGGS estimates local visitation density from a reference set sampled uniformly from the retained replay buffer. For a query latent z, let $\delta_1(z), \ldots, \delta_k(z)$ denote its k nearest-neighbor distances and define

$$\bar{\delta}_k(z) = \frac{1}{k} \sum_{j=1}^{k} \delta_j(z).$$

Let n_{eff} denote the effective reference-set size after excluding a self-match when present. The density estimate is

$$\hat{m}(z) = \frac{k}{n_{\text{eff}} v_d \max\left(\bar{\delta}_k(z), \epsilon\right)^d}, \tag{21}$$

where v_d is the volume of the d-dimensional unit ball. We use $k = 20$, $\epsilon = 10^{-12}$, and a reference set of up to $16{,}384$ states sampled from replay.

Define the log-density term

$$V(z) := \log(\epsilon + \hat{m}(z)). \tag{22}$$

The corresponding under-coverage score is $-V(z)$. A new reference set is sampled from replay at each collection cycle.

B.2 Candidate Goal Construction and Joint Goal-Set Selection

At each collection cycle, let N_R denote the number of transitions currently retained in replay. The recent window contains the most recent

$$N_{\text{recent}} = \min\{N_R, \max(512, \lceil 0.2 N_R \rceil)\}$$

transitions. EGGS samples 512 candidate goals without replacement from this window and 512 from its disjoint complement. Their embeddings and negative log-density values are evaluated under the current representation and remain fixed during goal selection.

Let $(g_1, \ldots, g_K)$ denote the selected goals and

$$Z = (z_1, \ldots, z_K), \qquad z_i = \phi_t(g_i)$$

their latent embeddings.

The implementation minimizes the energy

$$E(Z) = \lambda_V \frac{1}{K} \sum_{i=1}^{K} V(z_i) - \lambda_I \widehat{I}(Z), \tag{23}$$

where V is the log-density term defined in Eq. (22). With $\lambda_V = 1$, minimizing Eq. (23) is the minimization-form implementation of the Gaussian goal-set objective in Sec. 3.3, since the pointwise under-coverage utility is $-V(z)$. Unless otherwise stated, the reported AntMaze and locomotion runs use $K = 8$, $\lambda_V = 1$, $\lambda_I = 2$, and $\sigma_I = 30$. Since the latent output is not normalized, σ_I is expressed in the learned latent coordinates.

The Gaussian mutual-information term is evaluated with a Monte Carlo estimator. Let $M_{\mathrm{MC}} = 32$ and sample

$$\epsilon_m \sim \mathcal{N}(0, I_d), \qquad m = 1, \ldots, M_{\mathrm{MC}}.$$

For each goal embedding z_i, define

$$X_{im} = z_i + \sigma_I \epsilon_m.$$

The resulting mutual-information estimator is

$$\widehat{I}(Z) = \frac{1}{K M_{\mathrm{MC}}} \sum_{i=1}^{K} \sum_{m=1}^{M_{\mathrm{MC}}} \left[-\frac{\|\epsilon_m\|_2^2}{2} - \log \sum_{j=1}^{K} \exp\left(-\frac{\|z_i + \sigma_I \epsilon_m - z_j\|_2^2}{2\sigma_I^2} \right) + \log K \right]. \tag{24}$$

EGGS approximately optimizes Eq. (23) using multi-start discrete coordinate descent with eight initializations and at most 24 sweeps per restart. Selected candidate indices are constrained to be distinct.

B.3 Go-Explore Collection and Training

Each selected goal g_i is used in a fixed-horizon Go-Explore rollout. After the goal set is selected, the representation and policy parameters remain unchanged until the complete collection batch has been gathered.

Go policy. Go uses the Euclidean temporal distance

$$d_\phi(s, g) = \|\phi(s) - \phi(g)\|_2$$

and computes the transition reward as

$$r_{\mathrm{Go}}(s, s'; g) = d_\phi(s, g) - d_\phi(s', g). \tag{25}$$

Go is trained from the raw state and goal observations $[s_{\mathrm{raw}}, g_{\mathrm{raw}}]$ using this latent-distance progress reward.

For AntMaze, the total rollout horizon is 300 with $T_{\mathrm{go}} = 270$ and $T_E = 30$, while the locomotion environments use $T_{\mathrm{go}} = 170$ and $T_E = 30$ within a total horizon of 200. The simulator state at the end of Go is used directly as the initial state for Explore, and each Explore phase terminates after its fixed T_E-step horizon.

Explore policy. Goal selection and Explore both use replay-based kNN statistics to capture visitation under-coverage, but use different quantities for their respective roles. Goal selection uses the kNN density estimate defined in Appendix B.1, whereas Explore uses a normalized kNN-distance potential as its policy-learning signal. The Explore actor and critic take only the raw state observation s_{raw} as input; the goal and latent representation are not provided directly to the policy.

EGGS samples up to $65{,}536$ Explore reference states uniformly from retained replay. At each optimization round, these states are embedded using the representation at that round to form the reference set A used by the Explore kNN potential.

For a query latent z, let $\delta_1^E(z), \ldots, \delta_k^E(z)$ denote its k nearest-neighbor distances to A. The Explore potential is

$$N_A(z) = \log \left[1 + \frac{1}{k} \sum_{j=1}^{k} \max \left(\frac{\delta_j^E(z)}{\kappa_E} - 10^{-4}, 0 \right) \right], \tag{26}$$

with $k = 12$. Distances are normalized by a calibration scale κ_E, computed as the mean non-self kNN distance over up to $1{,}024$ sampled reference states.

The Explore transition reward is

$$r_E(s, s') = \alpha_E \left[\gamma_E N_A(\phi(s')) - N_A(\phi(s)) \right], \tag{27}$$

with

$$\gamma_E = 0.99, \qquad \alpha_E = 1.$$

This discounted-delta reward is return-equivalent, for a fixed Go endpoint and Explore potential, to maximizing the terminal Explore potential over the finite T_E-step Explore phase.

Collection and optimization schedule. At the beginning of each collection cycle, EGGS uses the current representation ϕ_t to construct candidate goals, evaluate negative log visitation density, and jointly select K exploratory goals using the Gaussian goal-set criterion in Appendix B.2. All Go-Explore trajectories for the selected goal set are then collected without parameter updates, after which the complete collection batch is appended to the replay buffer.

After replay insertion, EGGS constructs the Explore reference set and calibration scale from the updated replay buffer. During subsequent optimization, the representation and policies are updated from replay, and the latent-dependent quantities used by the Go and Explore rewards are recomputed using the representation at each optimization round. The next collection cycle uses the updated replay buffer and representation ϕ_{t+1}.

Algorithm 1 EGGS

Require: Replay buffer $\mathcal{D}$, number of selected goals K, horizons T_{go}, T_E

for iteration $t = 0, 1, \ldots$ **do**
  Estimate visitation density under the current representation ϕ_t
  Construct candidate goal set $\mathcal{G}_t$
  Select goals $\mathbf{g}_t^\star = (g_1, \ldots, g_K)$ according to Eq. (3)
  for each selected goal g_i **do**
    Collect one Go-Explore rollout: Go toward g_i for T_{go} steps, then Explore for T_E steps from the realized endpoint
  end for
  Add collected transitions to $\mathcal{D}$
  Update the representation to ϕ_{t+1} and update the Go-Explore policies from $\mathcal{D}$
end for

B.4 Training and Reproducibility Details

We collect the remaining training details needed to reproduce the AntMaze and locomotion experiments, including representation sampling, optimization, replay initialization, input normalization, and software configuration. Goal-selection parameters and Go-Explore rollout semantics are specified in Appendices B.2 and B.3, respectively.

Representation-learning sampling. EGGS retains the temporal representation objective and dual formulation of TLDR (Bae et al., 2024), while biasing representation minibatch sampling toward less densely covered replay regions. We uniformly sample without replacement up to $65{,}536$ transitions from the currently resident replay buffer and encode their next states using the mean of the current online representation encoder. For each reference transition i, let $\bar{d}_i$ denote the mean Euclidean distance to its 20 nearest neighbors within this reference set after excluding the self-match. Representation minibatches are sampled from this reference set according to

$$p_i = \frac{\bar{d}_i^2 + \epsilon_{\text{rep}}}{\sum_j \left(\bar{d}_j^2 + \epsilon_{\text{rep}} \right)}, \tag{28}$$

where ϵ_{rep} is float64 machine epsilon. The reference set and sampling weights are first computed after the initial collection cycle and are subsequently refreshed every eight collection cycles, or earlier when replay-buffer overwrites invalidate cached entries.

Temporal representation objective. For a representation minibatch, let $\mathrm{sh}(i)$ denote a one-position cyclic shift of the minibatch indices and define $\tilde{s}_i := s'_{\mathrm{sh}(i)}$. The representation objective is

$$\begin{aligned} \Delta_i^\phi &= \max\left(\|\phi(s_i) - \phi(\tilde{s}_i)\|_2^2, 10^{-6}\right)^{1/2}, \\ c_i &= \min\left[1 - \frac{1}{d}\|\phi(s'_i) - \phi(s_i)\|_2^2, 0.001\right], \\ J_\phi &= -\frac{1}{B_{\text{rep}}}\sum_i \mathrm{softplus}(500 - \Delta_i^\phi; \beta = 0.01) + \lambda_{\text{dual}}\frac{1}{B_{\text{rep}}}\sum_i c_i. \end{aligned} \tag{29}$$

The encoder minimizes $-J_\phi$, while $\lambda_{\text{dual}} = \exp(\xi)$ is optimized using the TLDR dual objective $L_{\text{dual}} = \xi\,\mathbb{E}[c(s, s')]$. The remaining representation hyperparameters are reported in Table 4.

Optimization schedule and relabeling. Training uses 0.03125 optimization rounds per newly collected physical interaction. Each round updates the temporal representation and dual variable, followed by Explore and Go SAC updates. Go training uses same-trajectory future-goal relabeling with probability 0.8; otherwise, the original selected goal is retained.

Network and optimizer settings. Go and Explore use tanh-Gaussian SAC policies (Haarnoja et al., 2018). Architecture and optimization hyperparameters are summarized in Table 4.

Table 4: EGGS training hyperparameters for the AntMaze and locomotion benchmarks.

Hyperparameter	Value
Latent dimension d	4
Replay capacity	1,000,000
Policy replay minibatch size	1,024
Representation minibatch size B_{rep}	1,024
Representation hidden layers	1024×2, ReLU
Go actor / twin-Q hidden layers	1024×2, ReLU
Explore actor / twin-Q hidden layers	1024×2, ReLU
Representation optimizer	Adam
Representation learning rate	5×10^{-4}
Go actor/Q/temperature optimizer	Adam
Go actor/Q/temperature learning rate	10^{-4}
Explore actor/Q/temperature optimizer	Adam
Explore actor/Q/temperature learning rate	10^{-4}
Dual optimizer	Adam
Dual learning rate	10^{-4}
Go discount	0.97
Explore discount	0.99
Target-network Polyak coefficient	0.005
Optimization rounds / collected physical interaction	0.03125
Go future-goal relabel ratio	0.8
Initial dual multiplier $\lambda_{\text{dual},0}$	3,000
Dual slack	0.001

Initialization and input normalization. Fresh runs initialize replay with 1,600 transitions collected by a seeded uniform-random policy before learner updates begin.

Representation inputs use fixed environment-specific normalization statistics throughout training, while policy and critic inputs remain in their native observation coordinates.

Software environment. The experiments use Python 3.8.20, PyTorch 2.0.1, CUDA 11.8, and MuJoCo 2.2.1.

C Experimental Details and Additional Results

This section provides evaluation protocols and supporting quantitative results omitted from the main text for space. Algorithmic training details and EGGS hyperparameters are provided in Appendix B.4. All tables report mean ± sample standard deviation across independent training seeds unless stated otherwise.

C.1 Controlled 2-D Exploration

We use the controlled 2-D maze implementation released with EDL (Campos et al., 2020) and evaluate the Square-A, Square-B, Square-C, Square-D, Square-Tree, and Square-Large layouts used in ExDM (Ying et al., 2026), with 10 independent seeds per method. Coverage and visitation-distribution metrics use the same fixed 0.2 task-space grid for all methods within each environment. The corresponding numbers of valid cells are 425 for Square-A, Square-B, and Square-C, 550 for Square-D, 775 for Square-Tree, and 2,500 for Square-Large. After the common 4,000-transition random initialization, the online interaction budget is 20,000 steps for Square-A, Square-B, Square-C, Square-D, and Square-Tree, and 25,000 steps for Square-Large.

Coverage AUC is computed from per-interaction cumulative coverage in chronological environment-interaction order using the trapezoidal rule. The common 4,000-transition initialization defines coverage at online step zero and is excluded from the reported online interaction budget.

For EGGS, the controlled 2-D experiments use the same temporal representation ϕ with latent dimension $d = 4$, while setting $K = 4$, $\lambda_I = 4$, and $\sigma_I = 20$. Each collection cycle considers 1,024 candidate goals. The 2-D control policies use the official Stable-Baselines3 SAC implementation (Raffin et al., 2021; Haarnoja et al., 2018), whereas the AntMaze and locomotion experiments retain the custom SAC implementation of TLDR.

The visitation metrics are computed by post-processing the stored trajectories used for the coverage evaluation; no additional training, retuning, or seed selection is performed. Each run contributes one stored position per online environment interaction together with the common 4,000-transition random initialization tape, and all positions are mapped to the same fixed grid. Invalid positions are excluded from the occupancy counts. The final normalized visitation entropy $\bar{H}_{\text{vis}}(T)$ is computed independently for each seed and then aggregated across the 10 seeds.

For EGGS and the reimplemented baselines, hyperparameters are selected on Square-C using coverage AUC and then fixed across the remaining environments. ExDM uses its released 2-D configuration without environment-specific retuning. Table 5 reports final coverage.

Table 5: Final coverage (%) on the controlled 2-D environments over 10 seeds. Values are mean ± sample standard deviation. Bold and underlined entries denote the highest and second-highest unrounded means within each environment, respectively.

Method	Square-A	Square-B	Square-C	Square-D	Square-Tree	Square-Large
TLDR	93.7 ± 3.3	$\underline{99.1 \pm 1.8}$	74.9 ± 4.6	98.8 ± 0.6	92.2 ± 1.5	48.0 ± 3.1
METRA	80.5 ± 3.2	72.0 ± 5.7	55.2 ± 3.3	85.6 ± 2.6	78.5 ± 3.2	42.7 ± 2.1
RND	86.2 ± 1.7	77.0 ± 4.1	79.6 ± 5.7	87.2 ± 1.7	76.1 ± 5.0	42.0 ± 2.5
DIAYN	77.6 ± 6.3	75.3 ± 10.5	67.6 ± 10.5	85.8 ± 3.3	66.9 ± 5.6	35.0 ± 4.7
CIC	99.6 ± 0.7	96.8 ± 3.6	84.9 ± 3.0	99.3 ± 0.7	$\underline{96.0 \pm 3.9}$	48.9 ± 3.2
ExDM	$\underline{100.0 \pm 0.1}$	98.9 ± 2.1	$\underline{88.9 \pm 7.2}$	$\mathbf{99.9 \pm 0.2}$	94.6 ± 3.8	$\underline{57.9 \pm 3.5}$
EGGS	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\underline{99.8 \pm 0.2}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{65.6 \pm 3.9}$

Support-normalized evenness. Normalized visitation entropy reflects both the size of the visited support and the concentration of visitation within that support. To examine within-support concen-

tration separately, we additionally report the final support-normalized evenness

$$E_{\text{support}} = \frac{\exp(H_{\text{vis}}(T))}{|\mathcal{V}_T|}, \tag{30}$$

where $H_{\text{vis}}(T)$ is the unnormalized final visitation entropy and $\mathcal{V}_T$ is the set of valid cells visited by the end of the run. This quantity is used only to interpret the visitation-entropy results, since high evenness within a small support does not by itself indicate broad exploration.

Table 6: Final support-normalized evenness on the controlled 2-D environments. Values are mean $\pm$ sample standard deviation over 10 seeds. Bold and underlined entries denote the highest and second-highest unrounded means within each environment, respectively.

Method	Square-A	Square-B	Square-C	Square-D	Square-Tree	Square-Large
TLDR	0.237 ± 0.034	0.399 ± 0.037	0.376 ± 0.042	0.398 ± 0.028	0.375 ± 0.035	0.196 ± 0.020
METRA	0.162 ± 0.019	0.145 ± 0.017	0.180 ± 0.015	0.208 ± 0.010	0.202 ± 0.020	0.137 ± 0.016
RND	0.209 ± 0.021	0.210 ± 0.016	0.260 ± 0.031	0.216 ± 0.026	0.212 ± 0.033	0.169 ± 0.024
DIAYN	0.280 ± 0.057	0.278 ± 0.040	0.299 ± 0.031	0.314 ± 0.054	0.275 ± 0.054	0.180 ± 0.018
CIC	$\underline{0.662 \pm 0.012}$	$\underline{0.682 \pm 0.027}$	$\mathbf{0.732 \pm 0.014}$	$\mathbf{0.685 \pm 0.021}$	$\underline{0.643 \pm 0.017}$	0.359 ± 0.029
ExDM	0.640 ± 0.019	0.641 ± 0.024	0.628 ± 0.019	$\underline{0.641 \pm 0.020}$	0.524 ± 0.028	$\mathbf{0.420 \pm 0.032}$
EGGS	$\mathbf{0.717 \pm 0.065}$	$\mathbf{0.781 \pm 0.041}$	$\underline{0.710 \pm 0.047}$	0.567 ± 0.038	$\mathbf{0.691 \pm 0.022}$	$\underline{0.401 \pm 0.041}$

Table 6 shows that EGGS has the highest mean support-normalized evenness in Square-A, Square-B, and Square-Tree, the second-highest mean in Square-C and Square-Large, and the third-highest mean in Square-D. Thus, EGGS's high normalized visitation entropy reflects more than support size alone in several environments, while within-support evenness remains environment dependent. The corresponding spatial visitation patterns are shown in Fig. 3.

C.2 AntMaze and Locomotion Evaluation

We evaluate AntMaze Large for 10M environment interactions, Medium for 5M, and U-Maze for 3M. The locomotion experiments on HalfCheetah, Hopper, and Walker2d each use 5M environment interactions. All high-dimensional results are aggregated over five independent training seeds per method. Within each environment, compared methods use the same environment-interaction budget and collection-rollout horizon: 300 steps for AntMaze and 200 steps for locomotion. Figure 4 shows the AntMaze and locomotion environments.

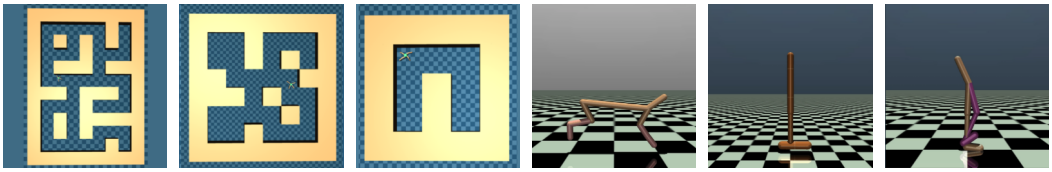


Figure 4: Evaluation environments used in the AntMaze and locomotion experiments.

AntMaze coverage is the fraction of valid planar-position cells visited during pretraining. For locomotion, we use continuous positional coverage based on the global forward position x. At interaction t, define

$$x_{\max}(t) := \max_{\tau \le t} x_\tau, \qquad x_{\min}(t) := \min_{\tau \le t} x_\tau,$$

and

$$C(t) = x_{\max}(t) - x_{\min}(t). \tag{31}$$

The extrema are accumulated over the full pretraining interaction history and are not reset at episode boundaries. Locomotion coverage is reported in each environment's native position units without

cross-environment normalization, so comparisons are made only among methods within the same environment.

EGGS selector hyperparameters were selected on AntMaze Large and fixed for Medium and U-Maze. The reported locomotion results use the same selector configuration. Tables 7 and 8 report time-normalized coverage AUC and final coverage, respectively. The AUC values summarize the coverage curves in Fig. 2; final coverage reports the value at the end of pretraining.

Table 7: Exploration AUC on the high-dimensional benchmarks. AntMaze reports the time-normalized AUC of maze-cell coverage (%), while locomotion reports the AUC of continuous positional coverage. Values are mean ± sample standard deviation over five independent training seeds; higher is better.

	AntMaze coverage AUC (%)			Locomotion coverage AUC		
Method	Large	Medium	U-Maze	HalfCheetah	Hopper	Walker2d
TLDR	$\mathbf{50.3 \pm 3.6}$	67.1 ± 6.2	82.2 ± 1.3	83.75 ± 2.10	2.74 ± 0.29	5.20 ± 0.06
METRA	22.9 ± 5.1	23.6 ± 4.5	29.2 ± 5.9	$\mathbf{96.74 \pm 6.30}$	2.49 ± 0.70	4.92 ± 0.46
EGGS	45.0 ± 5.5	$\mathbf{74.5 \pm 2.0}$	$\mathbf{84.1 \pm 3.7}$	77.86 ± 6.05	$\mathbf{4.68 \pm 0.19}$	$\mathbf{7.22 \pm 0.38}$

Table 8: Final high-dimensional exploration coverage over five independent training seeds. AntMaze entries report the percentage of valid planar-position cells visited by the end of pretraining, whereas locomotion entries report continuous positional coverage in native position units. Values are mean ± sample standard deviation. Comparisons are made within each environment.

	AntMaze final coverage (%)			Locomotion positional coverage		
Method	Large	Medium	U-Maze	HalfCheetah	Hopper	Walker2d
TLDR	$\mathbf{72.8 \pm 7.1}$	95.5 ± 9.1	$\mathbf{100.0 \pm 0.0}$	123.24 ± 4.06	3.36 ± 0.50	6.63 ± 0.33
METRA	36.8 ± 5.3	46.0 ± 8.3	47.7 ± 16.1	$\mathbf{125.23 \pm 6.79}$	3.50 ± 0.76	6.60 ± 0.50
EGGS	71.8 ± 6.9	$\mathbf{98.8 \pm 0.8}$	$\mathbf{100.0 \pm 0.0}$	99.07 ± 6.84	$\mathbf{5.62 \pm 0.28}$	$\mathbf{9.38 \pm 0.51}$

Across AntMaze, EGGS has the highest mean coverage AUC on Medium and U-Maze and the highest mean final coverage on Medium; on Large, TLDR has the highest mean on both metrics, while EGGS and TLDR both reach full final coverage on U-Maze. In locomotion, EGGS has the highest mean coverage AUC and final coverage on Hopper and Walker2d, whereas METRA has the highest mean on HalfCheetah.

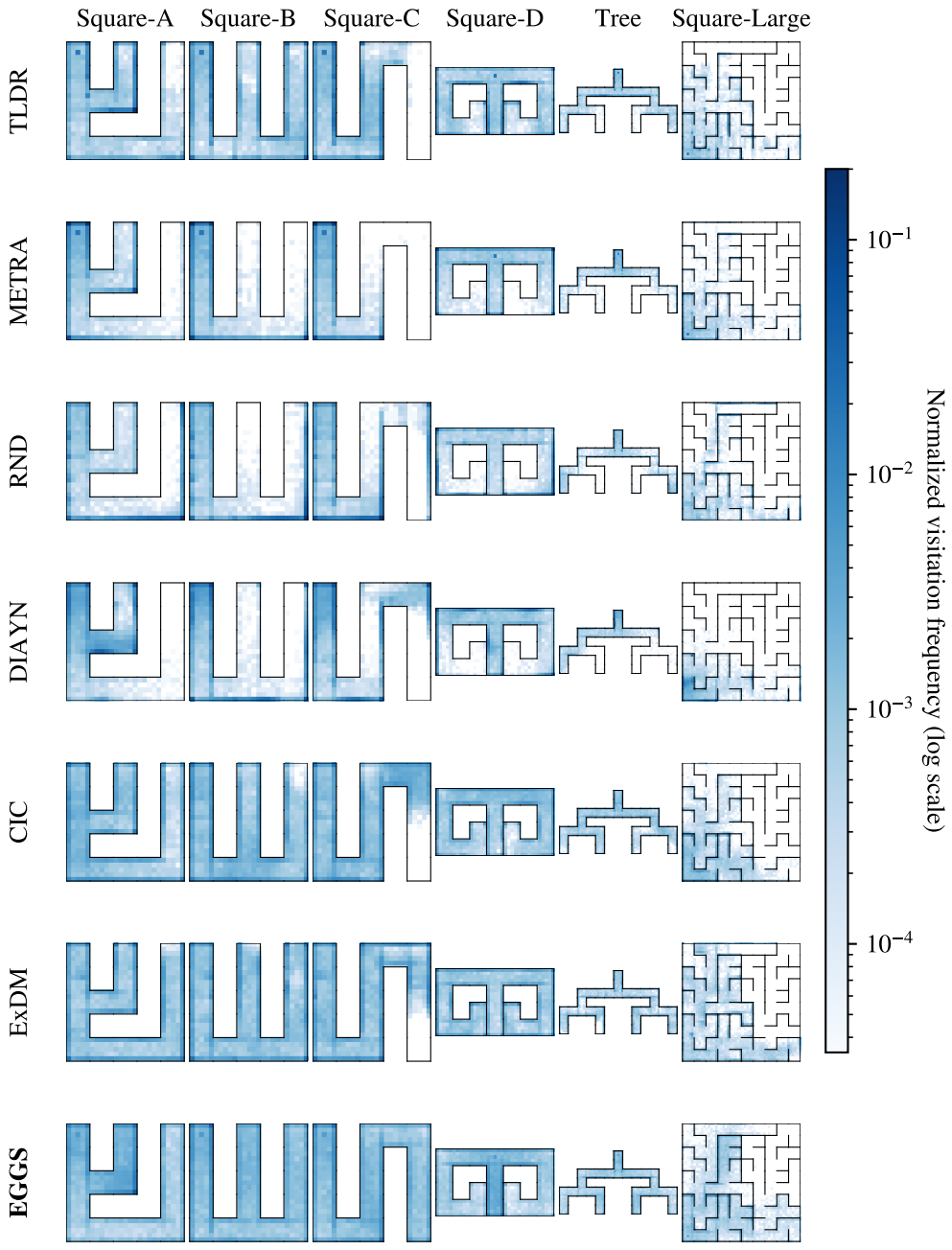


Figure 3: Cumulative visitation distributions in the controlled 2-D environments. Rows correspond to exploration methods and columns to environments. Color indicates normalized visitation frequency on a logarithmic scale. For each method-environment pair, we show the run whose final coverage is closest to the median across ten seeds, breaking ties by the smaller seed value.

C.3 Component Analysis and Sensitivity

We conduct the component analysis and goal-selection parameter sensitivity studies on AntMaze Large using three matched training seeds. These studies use a 5M-interaction budget. EGGS selector hyperparameters were selected separately on the full AntMaze Large training horizon and then held fixed for these analyses. All variants within each study use the same representation objective, representation-sampling scheme, network architecture, interaction budget, and evaluation protocol.

The component study compares Explore-only, single-goal Go-Explore, multi-goal selection without MI, and multi-goal selection with the Gaussian distinguishability term. Table 9 reports the component results and sensitivity studies, which vary goal-set size, distinguishability weight, and Gaussian scale one at a time.

Table 9: Component analysis and goal-selection parameter sensitivity on AntMaze Large over 5M interactions. Coverage AUC and final coverage are reported as mean ± sample standard deviation over three matched training seeds. Final coverage is the percentage of the 736 valid planar-position cells visited by the end of training. Higher is better. Identical configurations across study blocks reuse the same control runs.

Study	Variant	K	λ_I	σ_I	AUC (%) ↑	Final (%) ↑
Component analysis						
	Explore-only	–	–	–	22.4 ± 0.3	26.5 ± 1.2
	Single-goal	1	0	–	36.3 ± 2.1	50.0 ± 2.1
	Multi-goal, no MI	8	0	–	38.6 ± 1.7	53.1 ± 3.3
	Multi-goal + MI	8	2	30	38.4 ± 2.6	52.8 ± 2.9
Goal-set size sensitivity ($\lambda_I = 0$)						
	$K = 1$	1	0	–	36.3 ± 2.1	50.0 ± 2.1
	$K = 4$	4	0	–	39.5 ± 0.2	53.6 ± 0.8
	$K = 8$	8	0	–	38.6 ± 1.7	53.1 ± 3.3
Distinguishability-weight sensitivity ($K = 8, \sigma_I = 30$)						
	$\lambda_I = 1$	8	1	30	39.2 ± 1.4	55.6 ± 2.9
	$\lambda_I = 2$	8	2	30	38.4 ± 2.6	52.8 ± 2.9
	$\lambda_I = 4$	8	4	30	39.0 ± 1.0	54.3 ± 1.8
Gaussian-scale sensitivity ($K = 8, \lambda_I = 2$)						
	$\sigma_I = 15$	8	2	15	37.7 ± 3.1	52.4 ± 5.5
	$\sigma_I = 30$	8	2	30	38.4 ± 2.6	52.8 ± 2.9
	$\sigma_I = 60$	8	2	60	39.4 ± 0.5	55.3 ± 2.5

Controlled 2-D distinguishability ablation. We additionally isolate the effect of Gaussian distinguishability in the controlled 2-D environments using 10 paired seeds. The compared configurations use identical seed indices and differ only in whether the Gaussian distinguishability term is enabled. Table 10 reports final normalized visitation entropy and coverage AUC together with the paired difference between the two configurations.

The paired comparison shows negligible or mixed mean changes on Square-A–C. On Square-D, Square-Tree, and Square-Large, enabling Gaussian distinguishability yields positive mean changes in both final visitation entropy and coverage AUC, with the largest changes on Square-Large. On Square-Large, the paired difference is positive in eight of ten seeds for each metric.

Table 10: Controlled 2-D ablation of Gaussian distinguishability over 10 paired seeds. Absolute values are mean ± sample standard deviation across seeds. Δ is computed seed-wise as $+\mathrm{MI}$ minus no-MI and is reported as mean ± sample standard deviation across the 10 paired differences. Absolute values are reported as percentages, and paired differences are reported in percentage points.

	Final normalized visitation entropy (%)			Coverage AUC (%)		
Environment	No MI	+MI	Δ	No MI	+MI	Δ
Square-A	94.46 ± 1.44	94.45 ± 1.57	-0.01 ± 0.80	95.81 ± 1.23	95.80 ± 1.41	-0.01 ± 0.26
Square-B	95.84 ± 0.86	95.90 ± 0.87	$+0.06 \pm 0.49$	97.39 ± 1.00	97.33 ± 1.08	-0.06 ± 0.27
Square-C	94.82 ± 0.82	94.32 ± 1.12	-0.50 ± 0.55	94.04 ± 2.94	93.82 ± 3.44	-0.21 ± 0.60
Square-D	90.72 ± 1.19	90.95 ± 1.11	$+0.23 \pm 0.75$	96.43 ± 0.89	96.63 ± 1.09	$+0.20 \pm 0.66$
Square-Tree	93.92 ± 0.59	94.44 ± 0.47	$+0.52 \pm 0.77$	93.42 ± 1.08	94.02 ± 0.70	$+0.60 \pm 0.84$
Square-Large	81.77 ± 2.15	82.86 ± 1.98	$+1.09 \pm 1.41$	51.96 ± 3.81	54.66 ± 3.12	$+2.69 \pm 3.35$

C.4 Mechanism Diagnostic

We conduct a matched diagnostic on AntMaze Large to compare how pointwise utility, EGGS's Gaussian distinguishability criterion, and greedy farthest-point diversification affect the selected goal sets and the resulting Go-Explore trajectories. The diagnostic uses 6M-interaction EGGS checkpoints from five independently trained seeds,

$$\{17, 19, 29, 37, 43\}.$$

All repeated trials and rollout repetitions are averaged within each training seed before aggregation, so the statistical unit is the training seed.

Part A: Goal-set geometry. For each training seed, we freeze the checkpoint replay buffer, representation, normalizer, and Go-Explore policies and run 10 predetermined RNG trials. Within each trial, we sample $1{,}024$ candidate goals and $16{,}384$ density-reference states from the same replay snapshot. Candidate embeddings and pointwise under-coverage utilities are computed once and shared by all three selectors.

We compare three selectors with $K = 8$. Pointwise Top-K selects the eight candidates with highest pointwise under-coverage utility. EGGS uses Eq. (3) with $\lambda_I = 2$ and $\sigma_I = 30$. Greedy farthest-point selection (FPS) first selects the highest-utility candidate and then repeatedly selects the candidate with the largest minimum Euclidean distance in the frozen temporal representation to the goals already selected. All selectors enforce distinct candidate indices. No environment interaction, replay insertion, representation update, or policy update occurs during Part A.

Goal-set geometry is measured using the official AntMaze 1×1 planar grid only for diagnostic evaluation; this partition is not used by the continuous Gaussian selector. Let b denote the selected-goal cell counts. We report the categorical goal-set entropy $H(b/K)$, the number of distinct goal cells, and the normalized minimum latent separation

$$\frac{d_{\min}}{\sigma_I}, \qquad d_{\min} := \min_{i \neq j} \|\phi(g_i) - \phi(g_j)\|_2. \tag{32}$$

To characterize retained pointwise utility, each selected goal is assigned its percentile rank within the $1{,}024$ candidate utilities. Table 11 reports the selected-goal metrics, including mean and minimum utility percentiles and mean raw utility. For each metric, the 10 trials are first averaged within each training seed, after which we report the mean and sample standard deviation across the five seeds.

Table 11: Matched goal-set diagnostics on AntMaze Large at 6M interactions. All three selectors use identical candidate goals, density-reference states, and pointwise utilities within each trial. Values are mean ± sample standard deviation across five training seeds after averaging 10 matched trials within each seed. The planar grid is used only for diagnostic measurement and does not enter goal selection.

Selector	$H(b/K)$	Dist. cells	$d_{\min}/\sigma_I$	Mean pct.	Min. pct.	Raw utility
Pointwise Top-K	1.925 ± 0.104	7.14 ± 0.55	0.178 ± 0.091	99.66 ± 0.00	99.32 ± 0.00	16.10 ± 0.37
EGGS	2.031 ± 0.014	7.72 ± 0.08	0.732 ± 0.238	99.09 ± 0.16	96.66 ± 0.83	15.70 ± 0.43
FPS	2.079 ± 0.000	8.00 ± 0.00	2.826 ± 0.154	82.30 ± 2.02	50.50 ± 4.20	13.34 ± 0.40

Part B: Frozen one-batch collection. We next test whether the goal-set differences measured in Part A persist after Go-Explore collection. Part B uses the matched goal sets produced in Part A directly: the selectors are not rerun, and no goal set is selected based on rollout outcomes.

For each matched trial, each of the three selectors provides eight selected goals. The checkpoint Go and Explore policies and representation remain frozen throughout collection. Each selected goal is used for one 300-step Go-Explore trajectory, consisting of 270 Go steps followed by 30 Explore steps. Thus, each selector condition collects eight trajectories and $2{,}400$ environment interactions per batch. Each selected goal set is evaluated with three independent rollout repetitions.

Within each repetition, corresponding goal indices across the three selectors use the same reset, environment, and action RNG seeds. We additionally verify that the corresponding rollouts begin from identical initial observations. No optimizer, representation, policy, or replay update occurs during the diagnostic collection, and the policy and encoder state hashes are verified to remain unchanged before and after the rollouts.

Across five training seeds, 10 trials, three selectors, three rollout repetitions, and eight 300-step trajectories per selector and repetition, Part B contains $1{,}080{,}000$ diagnostic environment interactions.

All trajectory-level cell metrics use the same official AntMaze 1×1 planar grid as in Part A. Let S_i denote the set of cells visited by the ith trajectory in a diagnostic batch, and define the batch visitation set as

$$S_{\text{batch}} := \bigcup_{i=1}^{K} S_i.$$

Trajectory redundancy is measured by the mean pairwise Jaccard overlap

$$J_{\text{pair}} = \frac{2}{K(K-1)} \sum_{i<j} \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \tag{33}$$

and union efficiency

$$E_{\text{union}} = \frac{\left|\bigcup_{i=1}^{K} S_i\right|}{\sum_{i=1}^{K} |S_i|}. \tag{34}$$

We additionally report the total number of distinct cells visited by the batch, $|S_{\text{batch}}|$.

For each training seed, the three rollout repetitions are first averaged within each matched trial, and the resulting 10 trial values are then averaged within the seed. Table 12 reports mean ± sample standard deviation across the five training seeds.

Pointwise Top-K selection retains the highest under-coverage utility but produces the least diverse goal sets and the greatest trajectory overlap. FPS provides the opposite reference: it greedily prioritizes geometric separation and yields the lowest trajectory overlap, but substantially lowers the utility of the selected goals. EGGS lies between these extremes, retaining a mean utility percentile above 99 while substantially increasing goal-set diversity and reducing trajectory redundancy relative to pointwise selection. This pattern illustrates the utility–diversity tension examined by the finite-cell analysis without treating pure geometric separation as the sole objective.

Table 12: Frozen one-batch trajectory diagnostics for the three selector outputs from Part A at 6M interactions. Values are mean ± sample standard deviation across five training seeds after averaging three rollout repetitions and 10 matched trials within each seed.

Selector	Pairwise Jaccard ↓	Union efficiency ↑	Distinct visited cells ↑
Pointwise Top-K	0.231 ± 0.044	0.410 ± 0.049	111.63 ± 9.49
EGGS	0.134 ± 0.004	0.560 ± 0.020	130.39 ± 5.49
FPS	0.111 ± 0.002	0.627 ± 0.010	136.75 ± 7.09

C.5 Downstream Evaluation

The downstream evaluations test whether behaviors learned during reward-free pretraining remain useful when the pretrained representation and low-level controller are frozen. Each downstream run uses the final checkpoint of its corresponding pretraining run; no intermediate checkpoint is selected according to downstream performance. No low-level fine-tuning is performed, and in sequential HRL only the high-level learner is updated. Each method is evaluated using five independently pretrained seeds under common downstream targets and evaluation budgets.

Zero-shot goal reaching. We evaluate zero-shot goal reaching on AntMaze Medium and Large without reward learning or policy adaptation. All methods use the same reset–goal pairs within each evaluation, stratified by maze geodesic distance. For Medium, the evaluation contains 25 target goals and three reset seeds per goal, yielding 75 episodes per pretrained run. For Large, it contains 45 target goals and three reset seeds per goal, yielding 135 episodes per pretrained run. Controllers requiring full-state goals receive a common goal state constructed by replacing only the root position of a fixed canonical observation with the target coordinate. EGGS and TLDR receive this state-space goal directly. For METRA, let s_g denote the canonical goal observation obtained by replacing the planar position of the fixed reference observation with the target coordinate, and let s_t denote the current observation. Using the frozen encoder mean ϕ, we construct the method-native command

$$z_t = \frac{\phi(s_g) - \phi(s_t)}{\|\phi(s_g) - \phi(s_t)\|_2}.$$

The current observation and z_t are then provided to the frozen METRA option policy. When the pretrained checkpoint contains an input normalizer, the same frozen normalizer is reused; no downstream normalizer is fitted.

Each episode is rolled out for 1,000 primitive environment steps. Let $q_t = (x_t, y_t)$ denote the agent's planar position at step t, and let g denote the target coordinate. An episode is counted as successful if

$$\min_{0 \le t \le T} \|q_t - g\|_2 < 0.5. \tag{35}$$

Thus, success requires the trajectory to enter the open radius-0.5 neighborhood of the target at least once. Rollouts are not terminated upon success.

We additionally report normalized geodesic progress,

$$P_{\text{geo}} = \frac{d_{\text{geo}}(s_0, g) - d_{\text{geo}}(s_T, g)}{d_{\text{geo}}(s_0, g)}, \tag{36}$$

where d_{geo} denotes maze geodesic distance and s_T is the terminal state of the full 1,000-step rollout. The reported zero-shot metrics are success rate (SR) and P_{geo}.

Sequential multi-goal HRL. We evaluate downstream composability on AntMaze Medium and Large using identically configured high-level learners over frozen method-specific low-level controllers. Each method trains a custom SMDP-SAC high-level policy whose input is

$$h_t = [s_t,\ g_{xy},\ d_t/50] \in \mathbb{R}^{32},$$

where s_t is the 29-dimensional AntMaze observation, g_{xy} is the current task goal, and d_t is the remaining goal deadline in primitive steps. The policy outputs a relative planar displacement

$$\Delta g_t \in [-4, 4]^2,$$

which is added to the current position and projected onto the valid maze region to define the low-level subgoal. Each episode lasts at most 200 primitive environment steps and contains up to four sequential goals. Each goal has a 50-step deadline, and the high-level controller issues commands at intervals of at most 25 primitive steps. The actor and each critic have two hidden layers of 256 units. SAC uses twin Q-functions, target Q-networks, and a learned entropy coefficient. SMDP targets use discount γ^τ with $\gamma = 0.99$, where τ is the number of primitive steps executed by the low-level controller. Training uses a sparse success reward of 2.5 and segment-local HER with $k = 4$ relabeled goals. The downstream reward, interaction budget, and high-level learner configuration are shared across methods.

The high-level learner is trained for 500,000 primitive environment interactions. We evaluate the deterministic high-level policy at

$$t \in \{0, 50{,}000, 100{,}000, \ldots, 450{,}000, 500{,}000\}.$$

At each checkpoint, performance is averaged over 100 evaluation episodes. Let $G(t) \in [0, 4]$ denote the resulting mean number of goals achieved per episode. We report $G(500{,}000)$ as the final goals achieved per episode and define

$$\text{GoalsAUC} = \frac{1}{500{,}000} \int_0^{500{,}000} G(t)\, dt. \tag{37}$$

The integral is evaluated with the trapezoidal rule over the 11 evaluation checkpoints. Equivalently, Goals AUC is the area under the goals-achieved-per-episode curve after normalizing the training-interaction axis to $[0, 1]$. Evaluation interactions are not included in the training-interaction axis. Figure 5 shows the HRL learning curves, and Table 13 reports the downstream results.

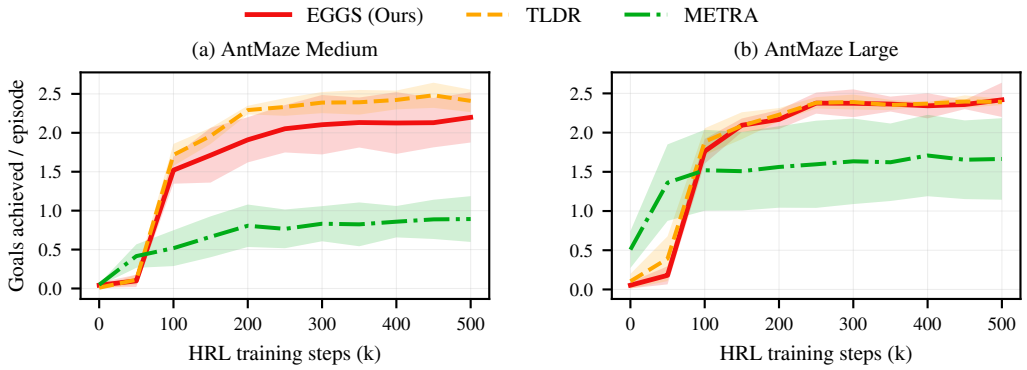


Figure 5: Sequential multi-goal HRL performance on AntMaze Medium and Large. Curves show the mean number of goals achieved per episode across five paired runs, where pretraining seed i is paired with downstream-training seed i; shaded regions indicate ± 1 SEM across the five pairs.

Table 13: Downstream evaluation on AntMaze Medium and Large over five independently pretrained seeds per method. Zero-shot goal reaching reports success rate (SR) and normalized geodesic progress P_{geo}. Sequential multi-goal HRL reports final goals achieved per episode and Goals AUC. Values are mean $\pm$ sample standard deviation across pretrained seeds. Higher is better for all metrics.

(a) Zero-shot goal reaching

Environment	Method	SR (%) ↑	P_{geo} ↑
Medium	TLDR	$\mathbf{83.7 \pm 5.8}$	0.927 ± 0.024
	METRA	20.8 ± 5.8	0.689 ± 0.079
	EGGS	83.5 ± 5.9	$\mathbf{0.932 \pm 0.025}$
Large	TLDR	48.1 ± 3.0	0.800 ± 0.027
	METRA	14.7 ± 5.1	0.650 ± 0.064
	EGGS	$\mathbf{53.0 \pm 6.7}$	$\mathbf{0.835 \pm 0.027}$

(b) Sequential multi-goal HRL

Environment	Method	Goals / episode ↑	Goals AUC ↑
Medium	TLDR	$\mathbf{2.410 \pm 0.145}$	$\mathbf{1.931 \pm 0.049}$
	METRA	0.892 ± 0.294	0.704 ± 0.221
	EGGS	2.198 ± 0.323	1.690 ± 0.250
Large	TLDR	2.392 ± 0.041	$\mathbf{1.973 \pm 0.033}$
	METRA	1.664 ± 0.520	1.525 ± 0.469
	EGGS	$\mathbf{2.418 \pm 0.219}$	1.925 ± 0.082