CROSSMAZE: BENCHMARKING ZERO-SHOT STRUCTURAL GENERALIZATION IN OFFLINE CONTINUOUS CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Policies trained on offline data are typically evaluated in familiar configurations represented in the training data, leaving their ability to transfer to unseen environments uncertain. We introduce **CrossMaze**, a benchmark for zero-shot structural generalization in offline continuous control. CrossMaze combines M16, a fixed diagnostic suite, with SeedMap, a procedural track for scaling training-layout diversity, in both PointMaze and AntMaze. We use CrossMaze to systematically evaluate eight imitation and offline reinforcement learning (RL) methods, including behavior cloning and offline goal-conditioned reinforcement learning (GCRL) across algorithm families and training-layout scales. As a strong reference baseline for CrossMaze, we introduce **PIDA**, a pretrained language-conditioned behavior-cloning policy that maps serialized maze layouts, states, and goals directly to continuous control actions. On M16, PIDA achieves held-out success rate of 69.8% on PointMaze and 72.7% on AntMaze, compared with 17.6% and 4.4% for the best evaluated non-PIDA baselines, respectively. Scaling experiments reveal that broader training-layout coverage yields uneven gains across algorithms and embodiments, while PIDA remains a strong reference with fewer layouts under the stated protocol. CrossMaze provides a controlled, reproducible testbed for measuring structural generalization in offline continuous control. Source code: `https://anonymous.4open.science/r/llm_offline-B54D/`

1 INTRODUCTION

Offline reinforcement learning (RL) aims to learn effective policies from a fixed dataset, making it particularly relevant when additional environment interaction is costly, unsafe, or simply unavailable. Existing benchmarks have driven substantial progress by exposing complementary challenges in this setting: D4RL exposed challenges arising from heterogeneous behavior and trajectory stitching (Fu et al., 2020), RL Unplugged expanded evaluation across diverse offline domains (Gulcehre et al., 2020), NeoRL emphasized conservative data collection and offline policy selection (Qin et al., 2022), and OGBench systematized long-horizon and goal-conditioned offline learning (Park et al., 2025). However, these benchmarks largely evaluate policies within the same environment structures represented in the offline data. Strong performance in these settings does not establish whether a learned policy can transfer to a Markov decision process (MDP) instance with an unseen transition structure. We study this extrapolation problem as *zero-shot structural generalization*.

Generalization across environments has been studied in several related settings. Procgen established a widely used protocol based on procedurally generated training and held-out levels (Cobbe et al., 2020). Subsequent work studies zero-shot generalization in offline Procgen (Mazoure et al., 2022) and found substantial train–test gaps across held-out Procgen levels and WebShop datasets (Mediratta et al., 2024). Recent theory further characterizes when zero-shot generalization from offline learning is possible (Wang et al., 2025; Weltevrede et al., 2026). Adjacent work studies generalization to unseen goals (Yang et al., 2023) and offline goal-conditioned learning across diverse tasks (Park et al., 2025). Our focus is complementary: rather than treating environment variation as an undifferentiated train–test shift, we isolate variation in the transition structure induced by the environment itself. This defines a structural axis of generalization that is difficult to disentangle in

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

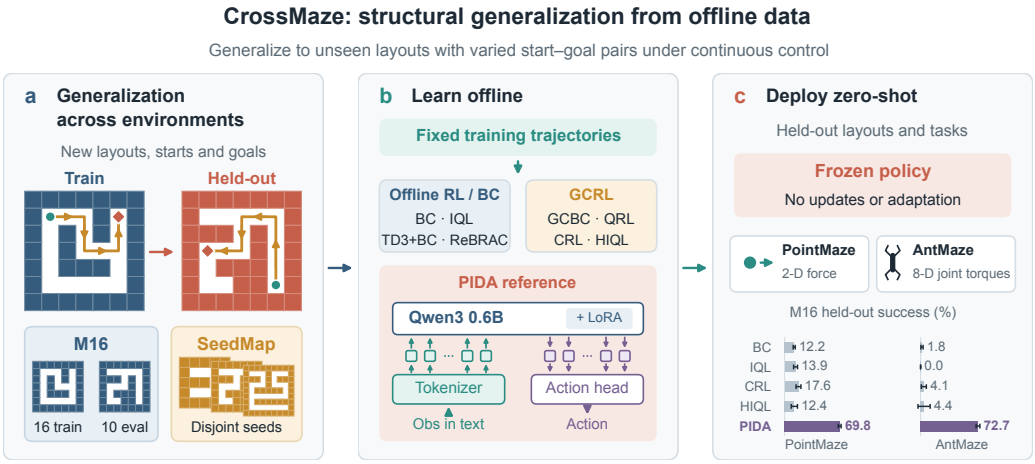


Figure 1: CrossMaze overview. **(a)** Layout and start–goal variation in M16 and SeedMap. Mazes and routes are illustrative; arrows connect circular starts to diamond goals. **(b)** Offline learning from fixed trajectories with shared state and goal inputs, excluding oracle routes or actions. PIDA is a pretrained language-conditioned behavior-cloning reference. **(c)** Frozen policies deploy, separately in PointMaze and AntMaze. Bars show M16 held-out success for PIDA and selected baselines.

broader multi-environment benchmarks, allowing *zero-shot structural generalization* to be measured directly in continuous control.

To study this setting systematically, we introduce **CrossMaze**, a benchmark for zero-shot structural generalization in offline continuous control. CrossMaze formulates layouts as contexts in a goal-conditioned contextual MDP, with disjoint training and evaluation layouts and zero-shot evaluation without parameter updates or additional training data. CrossMaze instantiates this protocol in PointMaze and AntMaze, where route selection is coupled with continuous control. Layout changes alter collision events, feasible transitions, and long-range connectivity. PointMaze requires managing velocity and inertia, while AntMaze adds contact-rich locomotion and multi-joint control. CrossMaze combines two complementary tracks: **M16**, a fixed diagnostic suite with 16 training and 10 held-out layouts, and **SeedMap**, a deterministic procedural track for scaling layout diversity. All policies receive the current layout through a canonical dynamic-map interface, without shortest paths or oracle actions. We provide fixed datasets for direct comparison and a configurable generator for broader data and scaling studies.

We evaluate representative behavior-cloning, value-based offline RL, and offline goal-conditioned RL (GCRL) baselines. We also introduce **PIDA** (Pre-trained Language Models for Interactive Decision and Action), a language-conditioned behavior-cloning policy that serves as a strong reference baseline for CrossMaze. PIDA uses a pretrained causal language model to map state, goal, physical descriptions, and layout information to continuous actions. On M16, these baselines generalize poorly to held-out layouts; PIDA substantially outperforms them but leaves ample room for improvement, highlighting CrossMaze's value for studying structural generalization. We further examine how structural generalization scales with training-layout diversity and whether PIDA's advantage arise from model scale or physical descriptions.

To summarize, we introduce a framework, benchmark, and reproducible protocol for studying zero-short structural generalization from offline data. We formalize zero-shot held-out-layout evaluation as a structural transition shift in a goal-conditioned contextual MDP, distinguishing layout context from episode goals and control embodiments. CrossMaze instantiates this framework through M16 and SeedMap in PointMaze and AntMaze, with a canonical dynamic-map interface and map-macro evaluation. With versioned datasets, configurable data generators, map and seed manifests, and unified rollout and result schemas, CrossMaze supports reproducible study of how offline policies generalize across unseen transition structures. The evaluation protocol compares imitation learning, offline RL, offline GCRL, and PIDA across training-layout scales, complemented by analyses of parameter count and PIDA's physical observation and action descriptions.

2 PROBLEM SETUP

2.1 OFFLINE GOAL-CONDITIONED CONTEXTUAL MARKOV DECISION PROCESSES

We formulate the problem as offline learning in a family of goal-conditioned contextual MDPs, combining the contextual MDP framework (Hallak et al., 2015; Kirk et al., 2023) with offline GCRL (Eysenbach et al., 2022; Park et al., 2025). Let $\mathcal{C}$ denote the context space, and let $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{G}$ denote shared ambient state, action, and goal-representation spaces. Each context $c \in \mathcal{C}$ indexes a finite-horizon goal-conditioned MDP

$$\mathcal{M}(c) = (\mathcal{S}, \mathcal{A}, \mathcal{G}_c, P_c, R_c, \nu_c, H), \qquad \mathcal{G}_c \subseteq \mathcal{G}, \tag{1}$$

where P_c is the transition kernel, H is the episode horizon, and $\mathcal{G}_c$ is the set of admissible goals in context c. The goal-conditioned reward is $R_c(s, a, s', g)$, equivalently $R_{c,g}(s, a, s')$, for $g \in \mathcal{G}_c$. The joint distribution ν_c specifies the initial state and goal of each episode.

The context c specifies the environment instance, including its transition dynamics and admissible goals, while g specifies the target within that instance. Given a benchmark-provided representation $\psi(c)$, the policy conditions on the current state, goal, and context representation. An episode is generated by

$$(s_0, g) \sim \nu_c, \qquad a_t \sim \pi(\cdot \mid s_t, g, \psi(c)), \qquad s_{t+1} \sim P_c(\cdot \mid s_t, a_t). \tag{2}$$

For each training context $c \in \mathcal{C}_{\text{train}}$, let $\mathcal{D}_c$ be a fixed dataset of previously collected trajectories. The offline training dataset and the held-out-context split satisfy

$$\mathcal{D} = \bigcup_{c \in \mathcal{C}_{\text{train}}} \mathcal{D}_c, \qquad \mathcal{C}_{\text{train}} \cap \mathcal{C}_{\text{eval}} = \emptyset. \tag{3}$$

Policy learning and model selection, including validation, hyperparameter tuning, and checkpoint selection, use only training-context data. For goal construction or relabeling, target states are drawn from training trajectories in the same context as the associated transition.

We evaluate *zero-shot generalization from offline data* by deploying the learned policy in contexts $c \in \mathcal{C}_{\text{eval}}$ according to Equation 2. The policy parameters remain fixed throughout evaluation, with no adaptation or in-context policy optimization based on evaluation transitions or rewards. Evaluation trajectories are used exclusively to estimate performance and are not incorporated into the offline dataset.

Let $Y(\pi, c, g, s_0) \in \{0, 1\}$ indicate whether policy π reaches goal g within H steps in context c from initial state s_0. We measure generalization by the expected success rate, averaged uniformly over evaluation contexts:

$$J_{\text{eval}}(\pi) = \frac{1}{|\mathcal{C}_{\text{eval}}|} \sum_{c \in \mathcal{C}_{\text{eval}}} \mathbb{E}_{(s_0, g) \sim \nu_c} \left[Y(\pi, c, g, s_0) \right], \tag{4}$$

For evaluation contexts, ν_c is a fixed start–goal distribution specified by the benchmark. In CrossMaze, each context corresponds to a layout, yielding a map-macro average in which every layout contributes equally, independently of its size or the number of evaluation episodes assigned to it.

2.2 STRUCTURAL GENERALIZATION ACROSS LAYOUT CONTEXTS

Under the formulation in Section 2.1, CrossMaze uses maze layout L as context $c = L$ and evaluates PointMaze and AntMaze separately, each with a fixed embodiment. Within each track, layouts share ambient spaces $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{G}$ but define distinct instances through P_L, $\mathcal{G}_L$, R_L, and ν_L. Each layout induces a grid graph $G_L = (V_L, E_L)$ over free cells. Changing obstacles can add or remove vertices and edges, alter collision outcomes, and change feasible continuous trajectories. The resulting change in P_L constitutes a *structural transition shift*. Closing a passage, for example, can invalidate a successful route without changing the start or goal coordinates.

Layouts also constrain admissible goals $\mathcal{G}_L$ and start–goal distributions ν_L, but need not use different reward rules. A shared goal-reaching rule $R_L(s, a, s', g) = r(s', g)$ on admissible goals remains compatible with structural transition shift because changing P_L alters the trajectories that yield

reward. CrossMaze thus targets *zero-shot structural generalization*: learning from offline training-layout data to reach goals in disjoint evaluation layouts given their context representations $\psi(L)$, without policy updates or additional training data.

This evaluation axis differs from three related settings in specific ways. *Unseen-goal generalization in a fixed environment*, as studied by GOAT (Yang et al., 2023), varies goals or their sampling distribution with a fixed context and transition kernel. CrossMaze also requires transfer across layout-dependent kernels. This distinction concerns evaluation protocols, not the learning capabilities of goal-conditioned algorithms. *Physical-parameter generalization*, exemplified by CARL's variations in mass, friction, and gravity (Benjamins et al., 2023), changes P_c through physical properties. CrossMaze varies obstacles and route connectivity within a fixed control embodiment. Both fit the contextual MDP formalism but differ in how transitions change. *Offline data-coverage generalization within a fixed MDP*, including trajectory stitching in D4RL (Fu et al., 2020), requires acting beyond the behavior data's state–action distribution while keeping the transition kernel and task fixed. CrossMaze retains offline learning constraints while evaluating layouts whose transition kernels are absent from the training data.

These distinctions make layout the unit of generalization, with disjoint training and evaluation layouts and performance aggregated across layouts by Equation 4. The next section presents M16 and SeedMap, which instantiate this protocol and share a layout-information interface for continuous control.

3 The CrossMaze Benchmark

CrossMaze follows five design goals. Explicit layout maps marking agent and goal locations reduce confounds from visual appearance and partial observability. All methods receive these maps without oracle routes or actions. A compact fixed suite supports stable multi-seed comparisons and per-layout diagnosis, while deterministic generation enables controlled map-count scaling. PointMaze and AntMaze couple layout generalization with distinct continuous-control demands. Versioned maps, datasets, start–goal pairs, and evaluation settings support reproducibility.

M16: a fixed diagnostic track. M16 contains 16 training layouts and 10 held-out evaluation layouts per embodiment, with separate fixed splits for PointMaze and AntMaze. Its compact size supports multi-seed comparisons and per-layout diagnosis. Layout selection is based on structural coverage, independently of PIDA or baseline performance; controllers verify feasibility and collect offline trajectories.

Layouts are selected to cover a range of structural properties characterized by the CrossMaze topology-difficulty metric. Its weighted aggregate is

$$s_{\mathrm{raw}} = 100\,(0.05u_N + 0.25u_L + 0.15u_R + 0.25u_D + 0.20u_A + 0.05u_T + 0.05u_X)\,, \qquad (5)$$

where $u_N, u_L, u_R, u_D, u_A, u_T, u_X \in [0, 1]$ are transformed component scores for path count, mean path length, mean turn count, detour, grid area, three-way junction count, and four-way junction count, respectively. The coefficients specify their relative weights, which sum to one. Path statistics summarize the set of maximal canonical shortest paths. The reported difficulty q_{topo} is obtained from s_{raw} by a fixed monotone rescaling; path construction, component normalization, and all score transformations are specified in Appendix A. The metric supports layout stratification and interpretation, rather than predicting policy performance.

PointMaze training layouts span three difficulty bands: low (4.37–20.30), intermediate (44.41–58.22), and high (66.01–83.44). AntMaze uses lower, more concentrated layout difficulty to keep the combined demands of navigation and locomotion tractable for learning. Most AntMaze layouts score 41.89–50.50, with `umaze` and `medium` below this range and `ultra` highest at 58.14; there is no comparable three-tier stratification. For each embodiment, held-out layouts have a similar difficulty distribution to the training set while remaining structurally distinct. SeedMap complements this fixed diagnostic suite by scaling the number of training layouts.

SeedMap: a scalable procedural track. SeedMap uses a deterministic procedural generator to provide scalable layout diversity in PointMaze and AntMaze. It generates connected mazes with a tree topology on the grid graph. Layout size and corridor configuration vary with the seed; a fixed

generator configuration and seed reproduce the same layout. Individual layouts have no predefined difficulty tiers or target scores, with structural diversity arising from procedural generation. Training and evaluation maps are generated from disjoint seed sets. Generator details and reproducibility records are provided in Appendix B.

Offline datasets. CrossMaze provides fixed offline datasets for PointMaze and AntMaze, with successful, high-quality navigation trajectories as the main data regime. For each embodiment, M16 provides 5,000 complete trajectories per training layout, totaling 80,000 trajectories across 16 layouts. SeedMap provides four configurations, expressed as layouts $\times$ trajectories per layout: a smaller 32×250 dataset with 8,000 trajectories, and three larger datasets, 100×250, 500×50, and 1000×25, each containing 25,000 trajectories. All methods use the same fixed data within each experimental condition.

CrossMaze also provides map generators for M16 and SeedMap and an offline data generator, allowing users to construct additional layouts and collect trajectories to extend both layout coverage and dataset size.

4 PIDA: Pre-trained Language Models for Interactive Decision and Action

PIDA is designed as a strong, reproducible reference policy for CrossMaze, not as a new general-purpose offline RL algorithm. It combines established components, including a pretrained causal language model, low-rank adaptation, learned queries, a continuous action head, and behavior cloning. The reference recipe tests whether a pretrained sequence model can efficiently turn an explicitly described, previously unseen structure and a semantically named continuous state into control.

State–language serialization. At time t, PIDA converts the observation o_t into text as $x_t = \mathrm{Serialize}(o_t)$, where o_t denotes the current state, goal, and layout information available to all policies. The text x_t is tokenized and passed to the pretrained LLM backbone. We defer the detailed input template to Appendix E.

Pretrained backbone and parallel continuous decoder. Our reference backbone is the 0.6B-parameter Qwen3 causal language model (Yang et al., 2025). The pretrained weights are frozen and adapted with LoRA (Hu et al., 2022). The default configuration uses rank 16, scaling factor 32, and applies LoRA to the query, key, value, and output projections of self-attention. The continuous decoder described below is trained jointly with these adapters.

Let the tokenized prompt contain T embeddings of width h. PIDA appends K learned action query embeddings $Q \in \mathbb{R}^{K \times h}$, with $K = 2$ for PointMaze and $K = 8$ for AntMaze. Learned queries are a standard mechanism for eliciting task-specific outputs from shared representations (Carion et al., 2020); here they provide a fixed-cost continuous interface without generating action text. The prompt portion retains causal attention. Each action query can attend to every non-padding prompt token and to every other action query, so all action dimensions are computed in a single model forward pass.

The final-layer query states $H_t^Q \in \mathbb{R}^{K \times h}$ are concatenated and processed jointly by an MLPResNet head: layer normalization, a linear projection with SiLU activation, two residual MLP blocks, a final layer normalization, and an output projection. The head produces a latent mean $\mu_\theta(x_t) \in \mathbb{R}^{d_a}$. Although K equals the action dimension in the reference configurations, the implementation does not impose a one-query to one-action coordinate assignment; the head can use all query states for every output. This collective decoding is important for AntMaze, whose joint torques must be coordinated.

Squashed-Gaussian behavior cloning. PIDA parameterizes a diagonal tanh-squashed Gaussian, a standard bounded continuous-action construction (Haarnoja et al., 2018):

$$z_t \sim \mathcal{N}\big(\mu_\theta(x_t), \mathrm{diag}(\sigma^2)\big), \qquad a_t = \tanh(z_t). \tag{6}$$

The log standard deviation is a learned state-independent vector with one entry per action dimension. In the reference configuration it is initialized to -1 and clamped to $[-2.5, 0.5]$. This choice gives the policy a likelihood objective while keeping the state dependence concentrated in the language-conditioned mean.

For dataset action $a \in [-1, 1]^{d_a}$, let $\bar{a} = \mathrm{clip}(a, -1 + \epsilon, 1 - \epsilon)$ and $u = \mathrm{atanh}(\bar{a})$. PIDA minimizes the mean negative log-likelihood

$$\mathcal{L}_{\mathrm{PIDA}}(\theta, \sigma) = \mathbb{E}_{(x,a)\sim\mathcal{D}} \left[\frac{1}{d_a} \sum_{j=1}^{d_a} \ell_j(x, a) \right],$$
$$\ell_j(x, a) = \frac{1}{2} \left(\frac{u_j - \mu_{\theta,j}(x)}{\sigma_j} \right)^2 + \log \sigma_j + \frac{1}{2} \log(2\pi) + \log(1 - \bar{a}_j^2 + \epsilon). \tag{7}$$

The final term is the change-of-variables correction for the tanh transform. Training updates the LoRA adapters, learned action queries, MLPResNet head, and log-standard-deviation vector. There is no critic, Bellman target, online reward, or test-time fine-tuning. At evaluation, PIDA acts deterministically with $a_t = \tanh(\mu_\theta(x_t))$ and clips the result to the native environment bounds. Consequently, differences between PIDA and the other baselines should be interpreted as differences in policy representation, pretraining, capacity, and objective—not as a capacity-matched attribution to language pretraining alone.

5 Experiments

5.1 Experimental setup

Research questions. Experiments are organized around four pre-specified questions:

- **RQ1** How well do offline learners generalize to held-out M16 layouts?
- **RQ2** How does generalization scale with the number of training layouts?
- **RQ3** Is PIDA's generalization advantage explained by parameter size?
- **RQ4** Do physical observation descriptions and action descriptions improve PIDA?

Baselines. We compare BC, TD3+BC (Fujimoto & Gu, 2021), IQL (Kostrikov et al., 2022), and ReBRAC (Tarasov et al., 2023) as standard offline learners; GCBC, QRL (Wang et al., 2023), CRL (Eysenbach et al., 2022), and HIQL (Park et al., 2023) as offline GCRL methods; and PIDA as the pretrained reference policy (Section 4). We include GCRL methods based on prior evidence of structural generalization: HIQL reports 78.2% test success on Procgen Maze after offline training on 1,000 maze layouts (Park et al., 2023). Procgen Maze tests transfer to unseen layouts through discrete grid movement, without continuous physical control (Cobbe et al., 2020). CrossMaze tests whether these methods also generalize when navigation is coupled with continuous dynamics.

Common protocol and Metrics. Within each experimental condition, methods use the same offline data and receive the same state, goal, and layout information, represented numerically for the eight baselines and serialized as text for PIDA. All policies act in the native continuous action space and share evaluation layouts, start–goal pairs, and horizons. Hyperparameter tuning and checkpoint selection use only training-side data, following the zero-shot protocol in Section 2.1. Implementation and validation details are provided in Appendix C. We report training- and held-out-layout success using the map-macro average in Equation 4. M16 comparisons and map-count scaling results report the mean and standard deviation over 5 model seeds, averaging across layouts within each seed first.

5.2 M16 main comparison

Tables 1 and 2 report success rates on the M16 benchmark. Each entry is the mean and standard deviation over 5 seeds. The four named columns show benchmark layouts, while *Other Train* averages the 12 custom training layouts and *Held-out Avg.* is the equal-map average over held-out layouts.

Table 1: PointMaze success rates (%) on M16. Values are mean ± standard deviation over 5 seeds. Bold indicates the best mean in each column.

Algorithm	Open	U-Maze	Medium	Large	Other Train	Held-out Avg.
BC	90.0 ±3.5	97.3 ±3.1	**96.0 ±2.0**	**82.0 ±4.0**	50.5 ±1.0	12.2 ±1.3
TD3+BC	64.4 ±7.7	85.6 ±7.7	56.7 ±3.3	53.3 ±6.7	30.4 ±3.0	6.1 ±1.5
IQL	13.3 ±0.0	15.6 ±5.1	11.1 ±5.1	13.3 ±3.3	28.3 ±0.7	13.9 ±2.5
ReBRAC	67.8 ±10.7	86.7 ±6.7	45.6 ±11.7	45.6 ±10.2	24.0 ±1.4	13.5 ±1.3
GCBC	93.3 ±5.8	88.9 ±7.7	78.9 ±5.1	71.1 ±7.7	52.2 ±1.7	12.2 ±2.4
QRL	48.9 ±31.0	68.9 ±45.3	41.1 ±30.2	35.6 ±17.1	8.4 ±2.7	9.3 ±2.9
CRL	91.1 ±5.1	95.6 ±3.8	80.0 ±3.3	52.2 ±1.9	49.4 ±4.9	17.6 ±3.1
HIQL	56.7 ±20.3	58.9 ±27.8	57.8 ±8.4	47.8 ±1.9	63.2 ±1.8	12.4 ±3.9
PIDA	**97.7 ±2.3**	**100.0 ±0.0**	62.3 ±23.8	74.3 ±10.7	**76.8 ±4.8**	**69.8 ±1.8**

Table 2: AntMaze success rates (%) on M16. Values are mean ± standard deviation over 5 seeds. Bold indicates the best mean in each column.

Algorithm	U-Maze	Medium	Large	Ultra	Other Train	Held-out Avg.
BC	0.0 ±0.0	0.0 ±0.0	11.3 ±5.0	2.7 ±4.6	11.6 ±2.2	1.8 ±1.4
TD3+BC	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.6 ±0.6	3.0 ±3.4
IQL	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	2.0 ±1.1	0.0 ±0.0
ReBRAC	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.8 ±0.7	1.1 ±1.1
GCBC	0.0 ±0.0	0.0 ±0.0	10.0 ±3.3	4.4 ±3.8	11.4 ±0.7	2.2 ±3.8
QRL	0.0 ±0.0	0.0 ±0.0	6.7 ±3.3	2.2 ±1.9	6.8 ±4.3	0.0 ±0.0
CRL	0.0 ±0.0	0.0 ±0.0	34.4 ±24.1	10.0 ±6.7	18.0 ±2.2	4.1 ±3.2
HIQL	0.0 ±0.0	0.0 ±0.0	27.8 ±5.1	4.4 ±5.1	15.2 ±5.2	4.4 ±7.7
PIDA	**93.3 ±3.1**	**66.0 ±9.2**	**64.7 ±4.6**	**65.3 ±9.5**	**76.7 ±1.9**	**72.7 ±2.4**

Results. PIDA provides the strongest held-out-layout performance in both embodiments. On PointMaze, it reaches 69.8% held-out success, compared with 17.6% for the strongest non-PIDA method, CRL, a gap of 52.2 percentage points. PIDA also attains the highest average on the custom training layouts, Open, and U-Maze, while BC remains strongest on Medium and Large. This pattern separates fitting the official and training layouts from transferring to new structures: several numeric methods perform well on individual training layouts but remain below 18% on the held-out average.

The separation is larger on AntMaze. PIDA obtains 72.7% held-out success, whereas the best non-PIDA mean is 4.4% from HIQL, a difference of 68.2 percentage points. PIDA also leads on all four named benchmark layouts and the custom-training aggregate. Among the numeric methods, CRL and HIQL are the strongest on the Large layout and on average over the custom training layouts, but neither transfers reliably to held-out layouts. Together, these results answer RQ1 in the evaluated regime: standard offline RL and goal-conditioned methods exhibit a substantial structural generalization gap, while PIDA establishes a much stronger reference point for both low-dimensional and contact-rich continuous control.

5.3 RQ2: Scaling with the number of training layouts

We train PIDA on 1, 2, 4, 8, or 16 M16 layouts and the eight non-pretrained baselines on 100, 500, or 1000 SeedMap layouts. PIDA trained on 32 SeedMap layouts (PIDA-32) provides a reference on the same held-out seeds. M16 and SeedMap use separate layout distributions and test sets. Both tracks show an overall improvement in generalization with more training layouts.

PIDA scaling on M16. PointMaze success rises from $34.2 \pm 1.0\%$ with one layout to $60.3 \pm 2.9\%$ with two, reaching $68.6 \pm 1.5\%$ with 16. AntMaze stays below 7% through four layouts, then reaches

$54.9 \pm 1.2\%$ with eight and $67.2 \pm 2.3\%$ with 16. Gains slow beyond 2 PointMaze layouts and 8 AntMaze layouts, indicating diminishing returns from additional environments (Figure 2(a),(b)).

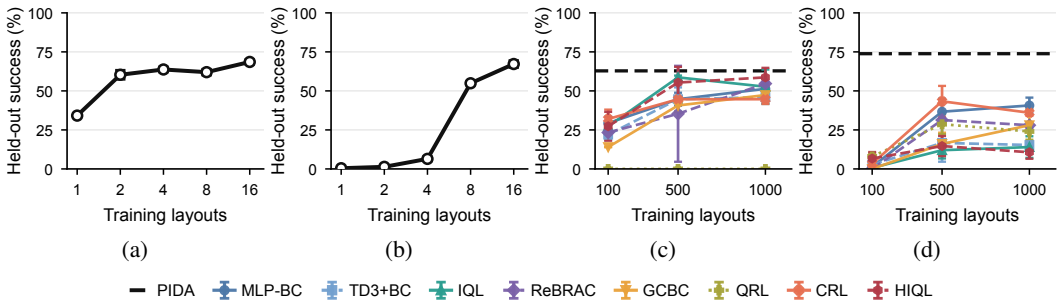


Figure 2: Held-out map-macro success versus training-layout count. (a,b) PIDA on M16 subsets. (c,d) Non-pretrained baselines on SeedMap; dashed horizontal lines mark PIDA-32 means. Each pair shows PointMaze then AntMaze. Markers and error bars show mean $\pm$ one standard deviation over 5 seeds.

Baseline scaling on SeedMap. Table 4 in Appendix D reports all baseline success rates at each training-layout count. Most of the eight baselines improve from 100 to 500 layouts, then show smaller gains or declines (Figure 2(c),(d)). On PointMaze, IQL rises from 27.3% at 100 layouts to $58.7 \pm 1.2\%$ at 500, then falls to $52.7 \pm 11.0\%$ at 1000. HIQL continues to improve beyond 500 layouts, but only from $55.3 \pm 10.1\%$ to $58.7 \pm 6.1\%$. AntMaze shows a similar pattern: CRL peaks at $43.3 \pm 9.9\%$ with 500 layouts before declining to $36.0 \pm 2.0\%$ at 1000, while MLP-BC improves modestly from $36.7 \pm 8.3\%$ to $40.7 \pm 5.0\%$. Thus, the benefit of broader training coverage largely weakens beyond 500 layouts, with further gains depending on the algorithm.

PIDA generalizes better with fewer training environments: 32 SeedMap layouts suffice to exceed every baseline mean at 100–1000 layouts in both embodiments. It achieves $62.8 \pm 1.5\%$ on PointMaze and $73.8 \pm 4.8\%$ on AntMaze, whose best baseline reaches $43.3 \pm 9.9\%$. These results highlight PIDA's ability to generalize from far fewer training layouts.

5.4 RQ3: Is PIDA's advantage explained by parameter size?

Both PIDA and MLP BC train only an actor through behavior cloning, making MLP BC a natural choice for testing whether greater policy capacity can close the generalization gap to PIDA. We scale MLP BC's parameter count by varying its width and depth on M16, holding inputs, offline data, evaluation episodes, and checkpoint selection fixed. The width sweep fixes depth at 4 layers; the depth sweep stacks 4-layer ResNet blocks at a fixed width of 512.

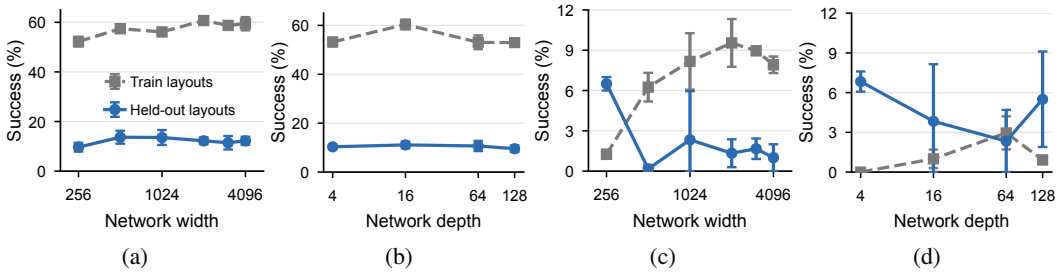


Figure 3: MLP BC capacity scaling on M16: width and depth sweeps for (a,b) PointMaze and (c,d) AntMaze. Blue solid and gray dashed curves show held-out and training success (mean $\pm$ one standard deviation over 5 seeds). Lower error bars are clipped at zero; vertical ranges differ by embodiment.

On PointMaze, training success generally rises as width increases from 256 to 2048, reaching $60.7 \pm 0.8\%$, then levels off at roughly 59–61% through width 4096 (Figure 3(a)). Held-out success improves from $9.7 \pm 1.8\%$ at width 256 to $13.7 \pm 2.6\%$ at 512, but subsequently trends downward rather than benefiting from further widening. Increasing depth from 4 to 16 layers raises training success from 53.2% to 60.4%; at 64 and 128 layers, it falls back to about 53%. Held-out success remains within 9.6–11.1%, showing little benefit from added depth (Figure 3(b)).

On AntMaze, training success increases steadily with width up to 2048, peaking at $9.5 \pm 1.8\%$, then declines to $7.9 \pm 0.6\%$ at 4096 (Figure 3(c)). Held-out success remains low and fluctuates across widths, with substantial seed variability at some settings and no consistent improvement. The depth sweep also yields low success: training performance rises from nearly zero at 4 layers to 3.0% at 64, then falls to 0.9% at 128. Held-out success fluctuates between 2.3% and 6.8%, again showing large relative seed variability and no stable scaling trend (Figure 3(d)).

All variants remain far below PIDA's 69.8% on PointMaze and 72.7% on AntMaze. Increasing MLP capacity therefore does not close the observed gap; this comparison cannot separate the effects of pretraining, architecture, representation, and optimization.

5.5 RQ4: Do physical observation descriptions and action descriptions improve PIDA?

We compare four prompts on AntMaze SeedMap-32. **Mini** omits descriptions of both observations and actions. **AD** adds the valid torque range and the ordering of the eight action dimensions. **Full** further explains the physical meaning of each observation field. **ROT** (Random Observation Tags) retains Full's action description but randomly permutes its observation labels, breaking their correspondence with the underlying physical quantities. All variants use the same numeric observations, dynamic maps, offline data, training budgets, and evaluation protocol.

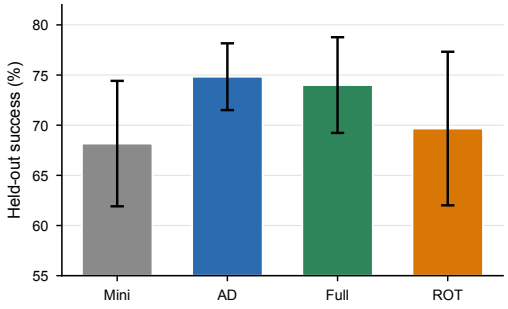


Figure 4: AntMaze SeedMap-32 prompt comparison. Bars show mean held-out map-macro success over 5 model seeds; error bars denote one standard deviation. The vertical axis starts at 55%.

Adding action descriptions improves held-out success from Mini's $68.2 \pm 6.3\%$ to AD's $74.8 \pm 3.3\%$ (Figure 4). Adding correct physical observation descriptions produces little change: Full achieves $74.0 \pm 4.8\%$, comparable to AD. Incorrect descriptions instead reduce mean success to 69.7% under ROT and increase the across-seed standard deviation from Full's 4.8 to 7.7 percentage points. Thus, action descriptions enhance performance, correct physical observation descriptions have little apparent effect, and incorrect ones weaken both performance and stability across seeds in this setting.

6 Conclusion

We introduced CrossMaze, a benchmark for zero-shot structural generalization in offline continuous control, combining the fixed M16 suite and procedural SeedMap track in PointMaze and AntMaze. We also introduced PIDA, a pretrained language-conditioned behavior-cloning policy, as a strong reference baseline. Our experiments reveal substantial generalization gaps among standard offline and goal-conditioned learners, with uneven gains from broader layout coverage. PIDA achieves higher held-out success with fewer training layouts under the stated protocol, while increasing MLP behavior-cloning capacity does not close the performance gap. Prompt comparisons suggest that action descriptions help, whereas physical observation descriptions offer little additional benefit in the evaluated setting. These findings motivate further study of how policy representations support structural transfer. CrossMaze provides a controlled, scalable framework for this research, with future extensions to imperfect maps and more varied offline data.

AI USE STATEMENT

Generative AI tools were used to assist with literature discovery, source-code inspection, and drafting. The authors are responsible for verifying the technical claims, citations, experiments, and final text.

REPRODUCIBILITY STATEMENT

The benchmark specification records environment and dataset versions, map and seed manifests, checksums, train/evaluation splits, start–goal lists, training configurations, checkpoint selection rules, and a machine-readable result schema. The final release will include the environment package, fixed datasets, data generators, baseline configurations, and unified evaluation entry points described in Sections 3 and 5.1.

REFERENCES

Carolin Benjamins, Theresa Eimer, Frederik Schubert, Aditya Mohan, Sebastian Döhler, André Biedenkapp, Bodo Rosenhahn, Frank Hutter, and Marius Lindauer. Contextualize me – the case for context in reinforcement learning. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=Y42xVBQusn.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229, 2020. doi: 10.1007/978-3-030-58452-8_13.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2048–2056, 2020. URL https://proceedings.mlr.press/v119/cobbe20a.html.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/e7663e974c4ee7a2b475a4775201ce1f-Abstract-Conference.html.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning, 2020. URL https://arxiv.org/abs/2004.07219.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/a8166da05c5a094f7dc03724b41886e5-Abstract.html.

Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Thomas Paine, Sergio Gómez, Konrad Zolna, Rishabh Agarwal, Josh Merel, Daniel Mankowitz, Cosmin Paduraru, Gabriel Dulac-Arnold, Jerry Li, Mohammad Norouzi, Matthew Hoffman, Nicolas Heess, and Nando de Freitas. RL Unplugged: A suite of benchmarks for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/51200d29d1fc15f5a71c1dab4bb54f7c-Abstract.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes, 2015. URL https://arxiv.org/abs/1502.02259.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

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

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Bogdan Mazoure, Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Improving zero-shot generalization in offline reinforcement learning using generalized similarity functions. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9fbdfded5c4d2969d889efc72f85c644-Abstract-Conference.html.

Ishita Mediratta, Qingfei You, Minqi Jiang, and Roberta Raileanu. The generalization gap in offline reinforcement learning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5c1ddd2e59df46fd2aa85c833b1b36ed-Abstract-Conference.html.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: Offline goal-conditioned RL with latent states as actions. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/6d7c4a0727e089ed6cdd3151cbe8d8ba-Abstract-Conference.html.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=M992mjgKzI.

Rong-Jun Qin, Xingyuan Zhang, Songyi Gao, Xiong-Hui Chen, Zewen Li, Weinan Zhang, and Yang Yu. NeoRL: A near real-world benchmark for offline reinforcement learning. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9cd828eb8dc81a84fb6bf89a94263e1b-Abstract-Datasets_and_Benchmarks.html.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/26cce1e512793f2072fd27c391e04652-Abstract-Conference.html.

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, 2023. URL https://proceedings.mlr.press/v202/wang23al.html.

Zhiyong Wang, Chen Yang, John C. S. Lui, and Dongruo Zhou. Provable zero-shot generalization in offline reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 65122–65143, 2025. URL https://proceedings.mlr.press/v267/wang25dx.html.

Max Weltevrede, Matthijs T. J. Spaan, and Wendelin Böhmer. Generalization in offline RL: The structure is more important than the amount of pessimism, 2026. URL https://arxiv.org/abs/2607.02288.

An Yang et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Rui Yang, Lin Yong, Xiaoteng Ma, Hao Hu, Chongjie Zhang, and Tong Zhang. What is essential for unseen goal generalization of offline goal-conditioned RL? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 39543–39571, 2023. URL https://proceedings.mlr.press/v202/yang23q.html.

A TOPOLOGY-DIFFICULTY METRIC IN M16

This appendix specifies the CrossMaze topology-difficulty metric used to characterize M16 and SeedMap layouts. The weighted aggregate is given in Equation 5; the definitions below specify its component scores and the transformation to the reported difficulty.

Path construction and structural statistics. Each maze is converted to an undirected four-neighbor graph over free cells. For every unordered pair of distinct free cells, a fixed-order breadth-first search selects one deterministic shortest path; a path and its reverse share a canonical representation. Paths that occur as contiguous subpaths of another canonical path are removed, leaving the set $\mathcal{P}$ of maximal canonical shortest paths. For $p \in \mathcal{P}$, let ℓ_p be its edge length, r_p its direction-change count, and $d_p = 1 - d_{\mathrm{Manhattan}}(p)/\ell_p$ its detour ratio, where $d_{\mathrm{Manhattan}}(p)$ is the Manhattan distance between its endpoints. Let $N = |\mathcal{P}|$, and let $\bar{\ell}$, $\bar{r}$, and $\bar{d}$ denote the respective means over $\mathcal{P}$. The remaining statistics are the full grid area A and the counts T and X of degree-three and degree-four corridor junctions whose diagonal neighbors are walls.

Component normalization and transformation. The structural statistics are normalized as

$$\begin{aligned} c_N &= \min\left(1, \frac{\log(1+N)}{\log 101}\right), & c_L &= \frac{\bar{\ell}}{\bar{\ell}+10}, \\ c_R &= \frac{\bar{r}}{\bar{r}+3}, & c_D &= \bar{d}\, c_L, \\ c_A &= \frac{A}{A+100}, & c_T &= \frac{T}{T+5}, \\ c_X &= \frac{X}{X+2}. \end{aligned} \tag{8}$$

The detour component is modulated by normalized path length. Each normalized component is then transformed independently:

$$u_i = f(c_i), \qquad f(x) = \frac{x^{1.5}}{x^{1.5} + (1-x)^{1.5}}, \qquad i \in \{N, L, R, D, A, T, X\}. \tag{9}$$

These u_i are the component scores used in Equation 5.

Reported score and auxiliary records. The weighted aggregate is mapped to the reported difficulty by

$$q_{\mathrm{topo}} = \frac{100}{1 + \exp[-(s_{\mathrm{raw}} - 45)/10]}. \tag{10}$$

The result is rounded to two decimals. Map size, free-cell count, and controller feasibility are recorded separately. Pair-level path difficulty, used to audit or stratify evaluation start–goal instances, is kept separate from this whole-map topology score.

B SEEDMAP GENERATION AND REPRODUCIBILITY

Deterministic layout generation. SeedMap defines a versioned deterministic mapping

$$(v, \kappa, z) \longmapsto L_z, \tag{11}$$

where v is the generator version, κ is a size specification, and z is a non-negative integer map seed. The version-1 generator constructs a bounded binary maze on an odd room–corridor lattice. It assigns stable hash-derived priorities to candidate edges and uses Kruskal's algorithm to carve a spanning tree, yielding a connected perfect maze. The default random-size specification samples square final matrix sizes from $\{5, 7, 9, 11, 13, 15\}$; fixed odd rectangular sizes are also supported. The same tuple produces the same matrix and SHA-256 map hash across runs.

Scaling protocol and release records. Training and evaluation seed sets are disjoint. Map-count sweeps record whether training sets are nested, since this determines whether larger training sets extend smaller ones or constitute separately sampled collections. Each release manifest stores the generator version, size specification, reward configuration, map seed, map hash, free-cell count, connectivity validation, and topology-difficulty statistics (Appendix A). We report the validity rate and the distributions of generated sizes and topology difficulties. The generator is not assumed to sample uniformly over maze layouts or to control each layout's difficulty individually.

C EXPERIMENTAL IMPLEMENTATION DETAILS

Baseline algorithms. The eight numerical baselines comprise standard offline learners and offline goal-conditioned methods. In the first group, **BC** is a supervised continuous policy that provides an imitation-learning reference for the high-quality demonstration data. **TD3+BC** augments TD3 with a behavior-cloning regularizer (Fujimoto & Gu, 2021). **IQL** learns values and extracts a policy without evaluating out-of-dataset actions (Kostrikov et al., 2022). **ReBRAC** is a behavior-regularized actor–critic baseline (Tarasov et al., 2023).

In the second group, **GCBC** performs behavior cloning after relabeling future states as goals. **QRL** learns a quasimetric goal-conditioned value function (Wang et al., 2023). **CRL** learns a goal-conditioned value representation through contrastive learning (Eysenbach et al., 2022). **HIQL** adds hierarchical policy extraction through latent subgoals (Park et al., 2023). These baselines are compared with **PIDA**, the pretrained language-conditioned BC policy described in Section 4.

Baseline implementations. BC, TD3+BC, IQL, and ReBRAC use a shared d3rlpy-based data and evaluation adapter. GCBC, QRL, CRL, and HIQL use a shared JAX/Flax implementation derived from the OGBench codebase.

Hyperparameter selection. We prioritize hyperparameters from each algorithm's original paper for related tasks, such as AntMaze or PointMaze. If these settings fail to yield an effective policy, we search around the published configuration. We select the best-performing configuration using training-side validation. For baselines without a single canonical source paper, such as BC, we use benchmark studies. Our BC reference configuration follows Mediratta et al. (2024). Table 3 reports the selected configurations.

M16 uses 5×10^5 optimizer updates for PointMaze; on AntMaze, BC uses 10^6 updates and the other methods use 3×10^6, with evaluation every 10^5 updates. The seven fixed-reference methods do not use early stopping. For SeedMap, seed 0 is trained for at least 10^6 and at most 5×10^6 updates, evaluated every 2.5×10^5 updates, and stopped after four training-map evaluations without an improvement of at least 0.02 in map-macro success; the other model seeds use the resulting seed-0 update budget. Held-out layouts are never used for this decision. Both backends normalize observations using training data.

Input representation. Numeric baselines receive the native numerical observation, desired goal, and dynamic-map matrix. Variable-sized maps are flattened in row-major order and padded to the largest shape within an embodiment, using -1 as a padding sentinel. PIDA receives the same observation, goal, and layout information in text form (Section 4).

Goal relabeling. For GCBC, QRL, CRL, and HIQL, relabeling preserves episode boundaries and never crosses layouts. Minibatches contain transitions from a single layout. Relabeled goals use complete future states under a fixed goal-semantics version, with map features recovered from the target state's original layout record. These rules apply to both M16 and SeedMap.

Validation and evaluation records. M16 uses an episode-level validation split of training-context trajectories. SeedMap validation partitions training-side map seeds before expanding trajectories, so a map cannot occur in both the fitting and validation subsets. Evaluation layouts are excluded from prompt, architecture, hyperparameter, and checkpoint selection. Per-layout and per-seed results are retained in the machine-readable outputs.

Table 3: Key hyperparameters of the numerical baselines. $d \times w$ denotes d hidden layers of width w. Actor and critic learning rates are equal where both are present.

Hyperparameter	Algorithm	Value
Network size	BC	4×1024
	TD3+BC, IQL	2×1024
	ReBRAC	3×1024
	GCBC, QRL, CRL, HIQL	3×1024
Activation	BC	SiLU
	TD3+BC, IQL, ReBRAC	ReLU
	GCBC, QRL, CRL, HIQL	GELU
Layer normalization	BC, GCBC, QRL, CRL, HIQL	Enabled
	ReBRAC	Critic only
	TD3+BC, IQL	Disabled
Batch size	BC, TD3+BC, IQL, ReBRAC	256
	GCBC, QRL, CRL, HIQL	1024
Learning rate	BC (M16 PointMaze)	1×10^{-4}
	ReBRAC	1×10^{-4}
	Others	3×10^{-4}
Discount factor γ	ReBRAC	0.999
	Others (except BC)	0.99
Target update rate τ	TD3+BC, IQL, ReBRAC, HIQL	0.005
Number of critics	TD3+BC, IQL, ReBRAC	2
Target policy noise	TD3+BC, ReBRAC	0.2
Target noise clip	TD3+BC, ReBRAC	0.5
Actor update delay	TD3+BC, ReBRAC	2
Actor-loss coefficient α	TD3+BC	2.5
	QRL	0.003
	CRL	0.03 (PointMaze); 0.1 (AntMaze)
	HIQL	3 (low- and high-level)
Actor BC coefficient β_{actor}	ReBRAC	0.002
Critic BC coefficient β_{critic}	ReBRAC	0.002
Expectile	IQL	0.9
	HIQL	0.7
Advantage temperature	IQL	10
Advantage weight cap	IQL	100
Reward shift	IQL	-1
Reward scale	ReBRAC	100
Latent dimension	QRL, CRL	512
IQE component dimension	QRL	8
QRL ϵ	QRL	0.05
Subgoal horizon	HIQL	25
Representation dimension	HIQL	10
Policy standard deviation	GCBC	1 (constant)
Policy loss	BC	Deterministic MSE (plain MLP)
	QRL	DDPG+BC
Value-goal sampling	QRL	Random
	CRL	Geometric future
	HIQL	Current / future / random: 0.2 / 0.5 / 0.3
Actor-goal sampling	GCBC, QRL, CRL, HIQL	Uniform future

D SeedMap baseline results

Table 4 provides the numerical results for the SeedMap baseline curves in Figure 2(c,d). We report held-out evaluation success, not training-layout success, for each algorithm trained on 100, 500, or 1000 layouts.

Table 4: SeedMap held-out map-macro success rates (%) for all eight baselines. n denotes the number of training layouts. Values are mean ± one sample standard deviation over three training seeds (0, 1, and 2), evaluated on the same 50 held-out layouts (map seeds 1001–1050) within each embodiment. Bold indicates the highest mean in each column.

	PointMaze			AntMaze		
Algorithm	$n = 100$	$n = 500$	$n = 1000$	$n = 100$	$n = 500$	$n = 1000$
MLP-BC	29.3 ± 4.2	44.7 ± 4.2	51.3 ± 3.1	1.3 ± 2.3	36.7 ± 8.3	$\mathbf{40.7 \pm 5.0}$
TD3+BC	22.0 ± 3.5	44.7 ± 7.6	46.0 ± 4.0	2.7 ± 4.6	16.7 ± 12.1	15.3 ± 1.2
IQL	27.3 ± 3.1	$\mathbf{58.7 \pm 1.2}$	52.7 ± 11.0	0.7 ± 1.2	12.0 ± 5.3	14.0 ± 6.9
ReBRAC	23.3 ± 1.2	35.3 ± 30.7	54.7 ± 1.2	0.0 ± 0.0	31.3 ± 1.2	28.0 ± 2.0
GCBC	14.0 ± 0.0	40.7 ± 4.2	47.3 ± 1.2	0.0 ± 0.0	16.0 ± 7.2	28.0 ± 10.0
QRL	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	$\mathbf{8.0 \pm 2.0}$	28.7 ± 6.4	24.0 ± 5.3
CRL	$\mathbf{32.0 \pm 6.0}$	44.7 ± 1.2	44.7 ± 3.1	3.3 ± 5.8	$\mathbf{43.3 \pm 9.9}$	36.0 ± 2.0
HIQL	27.3 ± 9.2	55.3 ± 10.1	$\mathbf{58.7 \pm 6.1}$	6.7 ± 4.2	14.7 ± 6.4	10.7 ± 4.2

E PIDA input template

CrossMaze provides the Map Visual as a numerical matrix. The numerical baselines receive this matrix directly, together with the numerical observation and goal. PIDA converts the same map information into the row-wise textual visualization shown below and serializes the current observation and goal as named numerical fields. In the visualization, `#` denotes a wall, `.` a free cell, `C` the current position, `G` the goal, and `S` a cell containing both current position and goal. The textual map is a rendering of the provided matrix, not an additional route or planning annotation.

The examples below show the input template for each embodiment. For readability, blue marks the physical observation explanations, orange marks the action description, and violet marks observation tags, including the variable names in the descriptions and the current observation. These colors are annotations in the paper; they are not part of the model input. The observation descriptions specify what the numerical state components represent, whereas the action description specifies the output dimensions and their range. The current-observation tags associate each displayed value with its state component. These examples use the Full prompt. In the AntMaze prompt comparison, Mini omits both description passages, AD includes the action description, and ROT changes the observation tags while retaining the numerical values and action description.

PointMaze example.

```
Visual (# = wall, . = open):
  # # # # # # # # # # # #
  # . . . . # . . G . . #
  # . # # . # . # . # . #
  # . . . . . . # . . . #
  # . # # # # . # # # . #
  # C . # . # . . . . . #
  # # . # . # . # . # # #
  # . . # . . . # . . . #
  # # # # # # # # # # # #

Observation semantics:
  x, y    - 2D position of the point mass in the maze
  vx, vy  - 2D velocity of the point mass
  gx, gy  - 2D goal position (desired_goal)

Predict the force components in the x and y directions,
each in the range [-1, 1].

Current observation (6-dimensional state vector):
  Position: (x=0.5674, y=1.8521)
  Velocity: (vx=0.0000, vy=0.0000)
  Goal:     (gx=2.5138, gy=-2.8695)
```

AntMaze example.

```
Visual maze (# = wall, . = open, C = current, G = goal,
             S = current and goal):
  # # # # # # # # # # # #
  # . . . . # . . G . . #
  # . # # . # . # . # . #
  # . . . . . . # . . . #
  # . # # # # . # # # . #
  # C . # . # . . . . . #
  # # . # . # . # . # # #
  # . . # . . . # . . . #
  # # # # # # # # # # # #

Observation semantics:
  x, y        - 2D torso position in the maze (achieved_goal)
  gx, gy      - 2D goal position (desired_goal)
  z, quat     - torso height and quaternion orientation [w, x, y, z]
  linear      - torso linear velocity [vx, vy, vz]
  angular     - torso angular velocity [wx, wy, wz]
  q, dq       - eight leg-joint angles and velocities in observation order

Predict the eight continuous joint torques in [-1, 1],
ordered as back-right hip/ankle, front-left hip/ankle,
front-right hip/ankle, and back-left hip/ankle.

Current observation:
  Position: (x=-19.76, y=-11.04)
  Goal:     (gx=21.00, gy=12.71)
  Torso:    z=0.58, quat=[0.98, 0.06, -0.07, -0.17]
  Velocity: linear=[1.86, 0.02, -1.36], angular=[-1.36, 0.75, -0.55]
  Joints:   q=[0.53, 0.52, -0.53, -0.52, 0.53, -0.52, 0.53, 0.55]
  JointVel: dq=[0.00, 0.04, 0.00, -0.01, -0.04, -0.01, -0.00, 1.70]
```