Action-Free Learning for Data-Efficient Offline Goal-Conditioned RL

Anonymous authors
Paper under double-blind review

Abstract

We study the data efficiency of offline reinforcement learning (RL) algorithms that leverage action-free data. A truly data-efficient algorithm should perform comparably whether trained on mostly action-free data with limited action-labeled transitions, or on an equivalent amount of fully action-labeled data. To evaluate how well current methods live up to this promise, we conduct controlled experiments using varying ratios of action-free and action-labeled data. Our empirical analysis yields two key findings. First, all evaluated methods struggle to match the performance achieved with a fully action-labeled dataset, revealing a bottleneck between the pretrained model's capacity and the fine-tuned policy's performance that widens as action labels become increasingly scarce. Second, we identify a counterintuitive trend: representation learning methods that provide stronger low-level policy initialization do not necessarily achieve higher success rates, suggesting effective planning, rather than action quality, is the primary driver of goal-reaching performance. Building on these insights, we develop an empirical recipe for data-efficient pretraining: (a) effective subgoal planning with value-based policy extraction is essential for strong goal-reaching performance; (b) representation learning methods should be used solely to initialize the low-level policy, not for planning; and (c) incorporating value functions consistently improves performance under action-limited settings. We instantiate this recipe in QSIA, a hierarchical method that substantially outperforms all evaluated baselines. Notably, fine-tuning QSIA with as little as 10% action-labeled data is generally sufficient to match the performance of training on a fully action-labeled dataset.

1 Introduction

A data-efficient offline RL algorithm must be able to considerably improve its performance by leveraging only a few interactions with the environment. However, in many real-world settings (e.g. robotics, healthcare, and traffic control) collecting samples can be expensive, slow, or dangerous. In this work, we are interested in offline RL algorithms that improve data efficiency by pretraining on action-free data to train performant policies from action-limited data. While prior work has shown that RL algorithms can leverage action-free data to train value functions (Laskin et al., 2020; Ma et al., 2023), planners (Seo et al., 2022; Black et al., 2024), value-based representations (Ghosh et al., 2023; Park et al., 2023), or dynamics-based representations (Edwards et al., 2019; Schmidt & Jiang, 2024), it remains unclear what makes these methods perform well when fine-tuned with action-limited data.

In principle, a truly data-efficient offline RL algorithm should be able to achieve performance comparable to training on fully action-labeled data, such that the agent's capabilities are bounded not by data quality alone, but by how effectively the algorithm exploits available data. Understanding how well current offline RL algorithms live up to this promise is important: it reveals whether existing approaches are already data-efficient, or whether further algorithmic advances are needed to more effectively leverage passive data.

To investigate this question, we study the effectiveness of current methods under varying ratios of action-free and action-labeled data. Specifically, we use robotics tasks from OGBench (Park et al., 2025a) across diverse domains, and partition the offline datasets by removing the majority of action labels from pretraining tra-

jectories, reserving the remaining action-labeled transitions for fine-tuning. Under these controlled settings, we evaluate the performance of state-of-the-art methods while systematically varying the data ratio.

We empirically study the behavior of existing methods under action-limited settings, focusing on two families of approaches that leverage action-free data: subgoal planning and representation learning. First, we show that all evaluated methods struggle to match the performance achieved with a fully action-labeled dataset, suggesting that a bottleneck exists between the pretrained model's capacity and the fine-tuned policy's performance, thus a gap that widens as action-labeled data becomes increasingly scarce. Second, we identify a counterintuitive trend: representation learning methods that provide stronger low-level policy initialization (e.g., latent action learning) tend to achieve lower success rates than methods that prioritize subgoal planning, despite producing higher action quality. This finding suggests that action quality alone is not the primary determinant of goal-reaching performance; rather, the effectiveness of high-level planning plays the greater role.

Building on these findings, we develop an empirical recipe for data-efficient learning from action-free data: (a) effective subgoal planning is necessary for strong goal-reaching performance; (b) representation learning should be used solely to initialize the low-level policy, not for planning; and (c) value functions are essential for improving success rates under action-limited settings. We demonstrate the effectiveness of this recipe by proposing **QSIA**, a hierarchical method that integrates these design principles. Using simple training objectives, QSIA demonstrates superior ability to leverage action-free data and fine-tune on limited action-labeled data. Notably, fine-tuning with only 10% action-labeled data is generally sufficient to match the performance of training on a fully action-labeled dataset.

Contributions.

1. Through controlled dataset partitioning, we empirically demonstrate that state-of-the-art methods capable of leveraging action-free data struggle significantly under scarce action-labeled settings.
2. We identify a counterintuitive finding: higher action quality does not necessarily lead to higher task success rates, and planning effectiveness is the primary driver of goal-reaching performance
3. Building on this insight, we motivate the use of action-quality-improving methods as an intermediate component that solely initializes the low-level policy, rather than also being responsible for planning.
4. We develop an empirical recipe derived from our analysis that improves data efficiency in action-limited settings.
5. We propose QSIA, which demonstrates strong performance under action-limited fine-tuning and retains most of the performance of a fully action-labeled baseline.

2 Related work

Offline Goal-Conditioned RL. Offline goal-conditioned RL combines the concepts of goal-conditioned RL (Kaelbling, 1993; Schaul et al., 2015) and offline RL (Lange et al., 2012; Levine et al., 2020) to train goal-reaching policies using previously collected unlabeled (reward-free) data. Previous studies on offline goal-conditioned RL have used several techniques, such as conservatism (Kumar et al., 2020), contrastive learning (Eysenbach et al., 2022), implicit $\{V, Q\}$-learning (Kostrikov et al., 2022; Park et al., 2023), and quasimetric geometry learning (Wang et al., 2023; Myers et al., 2024; 2026). Among these methods, we mainly consider hierarchical approaches, which enable action-free pretraining through representations, value functions, and high-level planners that can be used on downstream tasks (Park et al., 2023; 2026). In this work, we aim to study data-efficiency by using action-free data and limited action-labeled data, and develop a method that improves the performance and generalizability on action-limited settings.

Action-free Pretraining. Action-free and task-agnostic datasets are much more accessible and affordable than action-labeled demonstrations from experts. To enable agents to leverage action-free data, prior work has developed subgoal generators or planners (Black et al., 2024; Intelligence et al., 2026), value functions for subgoal selection (rejection sampling) (Mandlekar et al., 2020; Hatch et al., 2025; Park et al., 2026),

representations learning from goal-conditioned value functions (Ghosh et al., 2023; Bhateja et al., 2023; Park et al., 2023), and latent action models (Edwards et al., 2019; Schmidt & Jiang, 2024; Nikulin et al., 2025). Those methods based on subgoal generators have focused on the agent's planning ability, and incorporating rejection sampling improves the quality of the generated plans. This strategy enables the use of action-free, task-agnostic data with a high-level planner. While sensible, these approaches suffer when only a limited amount of action-labeled data is available, as shown in our experiments (Section 5.1).

Value-based representations learn representations from goal-conditioned value functions. Previous works have proposed several ways to use the representation, such as learning a state representation from the value function for downstream value learning (Ghosh et al., 2023; Bhateja et al., 2023), or learning a goal representation sufficient for control, which is then used to train a high-level planner (Park et al., 2023). While this last approach is effective in guiding policy and enables task-agnostic fine-tuning (Park et al., 2023; 2025a), it suffers when only scarce action-labeled data is available and is less effective at improving the action quality of low-level policies (Section 5.1).

Latent action learning (LAL) emerged as an approach that aims to recover action information from action-free data (Edwards et al., 2019; Schmidt & Jiang, 2024). Prior work has used latent actions to enable interaction with generated environments (Bruce et al., 2024), or demonstrated that using more complex architectures, such as transformers, enable pretraining on more diverse data (e.g. robotics datasets (Walke et al., 2023), human data) (Ye et al., 2025). Other works study the limitations of latent action models (LAMs), and find that training the Forward Dynamics Model (FDM) to reconstruct the next observation can harm the quality of the latent actions by encoding action-irrelevant information (Nikulin et al., 2025). Studies also find that discretized latent actions hinder the performance on continuous action spaces, which require more fine-grained representations, and address this issue by using a linear layer and the use of supervision with a small action-labeled dataset to improve the performance in manipulation tasks (Liang et al., 2025) and improve performance in the presence of distractors (Nikulin et al., 2025). These previous works mainly consider expert data for finetuning or short-horizon tasks, which limits the applicability of latent action models into more exploratory behaviors (Lynch et al., 2020) for finetuning or the horizon of the task it can solve. To address this, our method implements a subgoal planner that controls the LAM, and we prove that it is sufficient to enable task-agnostic finetuning and improve the success rate in long-horizon complex tasks. Recently, HiLAM (Kim et al., 2026) studied latent actions for long-horizon, language-conditioned robotic tasks, leveraging H-Net (Hwang et al., 2026), a complex hierarchical sequence model with dynamic chunking, to capture long-term temporal structure. In contrast, we focus on state–goal tasks and use a common MLP architecture across all methods to improve the long-horizon performance of latent actions by enabling more effective planning through rejection sampling and a subgoal generator.

Hierarchical RL Hierarchical RL (HRL) (Sutton et al., 1999; Stolle & Precup, 2002) aims to reduce the complexity of long-horizon problems by decomposing policies into distinct levels of abstraction. This principle is central to hierarchical GCRL (Schmidhuber, 1991; Kulkarni et al., 2016; Nachum et al., 2018; Levy et al., 2019), where a high-level policy outputs a desired future state (subgoal) that a low-level policy attempts to reach. Recent offline hierarchical GCRL algorithms (Gupta et al., 2020; Black et al., 2024; Hatch et al., 2025; Park et al., 2026) have adopted this structure to reduce the policy horizon for complex tasks. However, existing goal-conditioned hierarchies (Gupta et al., 2020; Park et al., 2026) predominantly rely on spatial subgoals as the sole interface between hierarchy levels, which implicitly assumes that sufficient action-labeled data is available for the low-level policy to learn effective goal-reaching behavior. We argue that this assumption does not hold in action-limited settings, where the scarcity of labeled transitions limits the ability of the low-level policy to accurately reproduce the actions required to reach a given subgoal. To address this limitation, we propose extending the hierarchical structure by incorporating an intermediate latent action model (LAM) that maps (state, subgoal) pairs into latent actions. Rather than relying solely on pretrained subgoal planners, the LAM recovers action-relevant features directly from action-free data, providing a richer initialization to the low-level policy. This design improves data efficiency in action-limited settings by enabling the low-level policy to better exploit the large amounts of available action-free data.

3 Preliminaries

Problem setting. We study the problem of offline GCRL defined by a Markov Decision Process (MDP) parameterized as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mu, p)$ (Puterman, 2014) and an unlabeled dataset $\mathcal{D}$ (without rewards), where $\mathcal{S}$ represents the state space, $\mathcal{A}$ represents the action space, $\mu \in \Delta(\mathcal{S})$ defines the initial state distribution and $p(s'|s,a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ denotes the transition dynamics function.

Our study focuses on a two stage process: Action-free pretraining stage and action-labeled training stage. First, we remove a ratio of the actions-labeled trajectories contained in $\mathcal{D}$ to create a passive dataset $\mathcal{D}_{\text{passive}}$ used for action-free pretraining. Second, the remaining trajectories in $\mathcal{D}$ are used for action-labeled training. Here, we assume that the entire state space $\mathcal{S}$ is identical to the goal space $\mathcal{G}$ and that trajectories are collected in a task-agnostic, unsupervised manner.

The objective is to learn from $\mathcal{D}_{\text{passive}} \cup \mathcal{D}$ an optimal policy $\pi(a|z)$ that maximizes the objective $J(\pi) = \mathbb{E}_{g\sim p(g), \tau\sim p^{\pi}(\tau)}[\sum_{t=0}^{\infty} \gamma^t r(s_t, g)]$, where z defines the latent action, γ is the discount factor, $p(g)$ denotes the goal distribution, and $p^{\pi}(\tau)$ is the trajectory distribution generated by the policy π.

Goal-conditioned value functions. This class of methods enable value functions to learn from unlabeled data (reward-free). Depending on the formulation of the method, it can be used for representation learning (Ghosh et al., 2023), policy extraction with advantage-weighted regression (AWR) (Peng et al., 2019), or as a subgoal classifier with rejection sampling (Mandlekar et al., 2020; Park et al., 2026). As an example, SHARSA (Park et al., 2026) fits a value function Q using n-step temporal difference (TD) learning with the following losses:

$$\mathcal{L}^V(V) = \mathbb{E}\Big[\text{BCE}(V(s_t, g), \bar{Q}(s_t, s_{t+n}, g))\Big], \tag{1}$$

$$\mathcal{L}^Q(Q) = \mathbb{E}\Big[\text{BCE}(Q(s_t, s_{t+n}, g), \sum_{i=0}^{n-1} \gamma^i r(s_{t+i}, g) + \gamma^n V(s_{t+n}, g))\Big], \tag{2}$$

where $r(s, g) = 1(s = g) - 1$ denotes the sparse goal-conditioned reward function, BCE defines the binary cross entropy loss, V is a value function, $\bar{Q}$ is the target Q network.

Hierarchical GCRL. Hierarchical goal-conditioned RL decomposes the goal-reaching problem into two policies: the high-level policy $\pi^h(w|s, g)$ acts as a planner which reduces the policy horizon by producing subgoals w that are spatially closer to the agent, and a low-level policy $\pi^{\ell}(a|s, w)$ which uses this closer future states to sample actions. As the high-level policy doesn't need any action information, it can be used for action-free pretraining Black et al. (2024). This class of methods can be combined with action-free value functions, it can either be for policy extraction with AWR Peng et al. (2019), reparameterized gradients (DDPG+BC) or rejection sampling, which picks the best subgoal based on the criterion of the Q function:

$$\pi^h(s, g) \stackrel{d}{=} \underset{w_1, \cdots, w_N : w_i \sim \pi^h_{\beta}(w|s,g)}{\arg\max} Q(s, w_i, g), \tag{3}$$

where $\stackrel{d}{=}$ denotes equality in distribution, N denotes the number of samples, and π^h_{β} defines a goal-conditioned high-level BC policy.

Latent action learning (LAL). Latent action learning methods aim to extract latent action information from state-only trajectories (e.g. videos) by learning a latent inverse dynamics model (IDM) via the objective of modeling forward dynamics. This class of methods often first learns these latent actions with the following objective:

$$z = \text{IDM}(s_t, s_{t+n}), \quad \mathcal{L}_{\text{LAL}} = \mathbb{E}_{(s_t, s_{t+1}, s_{t+n})\sim D}\Big[\parallel \text{FDM}(s_t, z) - s_{t+1} \parallel^2 \Big], \tag{4}$$

where FDM denotes the forward dynamics model, and s_{t+k} defines a state k steps away from s_t. Then, the IDM output can be used to train with behavioral cloning a high-level (latent action) policy π^h and a low-level policy π^{ℓ} to decode the latent action into the ground truth action:

$$J_{\pi^h_{\text{LAL}}} = \mathbb{E}_{(s_t, s_{t+n})\sim D, g\sim p^{\mathcal{D}}(g|\tau)}[\log(\pi^h(\text{IDM}(s_t, s_{t+n})|s, g))], \tag{5}$$

$$J_{\pi^\ell_{\text{LAL}}} = \mathbb{E}_{(s_t, a_t, s_{t+n}) \sim D}[\log(\pi^\ell(a_t | \text{IDM}(s_t, s_{t+n})))]. \tag{6}$$

Here, the high-level policy guides the low-level policy to reach the goal using the latent actions. In some LAL methods (Nikulin et al., 2025; Liang et al., 2025) the low-level policy objective $J_{\pi^\ell_{\text{LAL}}}$ also updates the weights of IDM.

4 Experimental setup

Tasks and datasets. We used 14 state-based tasks from the OGbench task suite (Park et al., 2025a), which include diverse robotic locomotion and manipulation tasks. In locomotion tasks, the agent aims to control a robotic body (a quadrupedal robot or a humanoid) to navigate a maze (antmaze, humanoidmaze) or dribble a soccer ball (antsoccer) to a desired goal. In manipulation tasks (cube, scene, puzzle), the agent learns to handle a robotic arm to arrange cubes, manipulate everyday objects, or solve combinatorial puzzles.

The standard OGBench datasets consist of "play" style (Lynch et al., 2020) task-agnostic trajectories that perform diverse random behaviors, such as random navigation through the maze, random dribbling of the ball, random pick-and-place sequences, or random button presses. For our experiments, we split these datasets into a large proportion of action-free data by removing action labels and the remaining trajectories as the action-labeled dataset.

Methods we evaluate. We evaluate four representative methods for action-free pretraining, spanning two broad categories:

Subgoal planning. We consider Hierarchical FBC (HGCFBC) (Park et al., 2026), a method that trains a hierarchical policy using flow-based behavioral cloning, where the high-level policy acts as a planner that directly predicts a raw future state as a subgoal. We also evaluate SHARSA (Park et al., 2026), a hierarchical method whose high-level planner predicts multiple candidate future states and incorporates a Q-function to apply rejection sampling, selecting the highest-value subgoal for the low-level policy to pursue.

Representation learning. We consider HIQL (Park et al., 2023), a hierarchical method in which the high-level policy predicts a subgoal representation that is learned jointly with the value function, enabling the agent to plan in a structured latent space. We also evaluate GCLAOM (see Appendix A for details), a goal-conditioned variant of the latent action learning method LAOM (Nikulin et al., 2025), which first trains an inverse dynamics model (IDM) to infer latent actions from state transitions, and subsequently trains a latent action model that conditions the low-level policy on these latent actions to guide execution.

5 How can we effectively use action-free pretraining techniques?

5.1 Empirical analysis 1: How do action-free pretraining techniques behave under action-limited settings?

In this empirical analysis, we evaluate four methods: Hierarchical FBC, a goal-conditioned variant of LAOM (GCLAOM), HIQL, and SHARSA. To assess the ability of these methods to learn from task-agnostic, action-free data, we use the OGBench datasets and conduct experiments across three action-limited settings, as well as the fully action-labeled dataset. Specifically, we vary the ratio of action-labeled to action-free data as follows: $2\%/98\%, 5\%/95\%, 10\%/90\%$, and $100\%/0\%$ (full supervision).

Figure 1 shows that, while all evaluated methods are capable of leveraging passive data to some degree, none of them achieves a success rate in the action-limited settings that is comparable to their performance under full supervision (100% action-labeled). Among the methods studied, three distinct behavioral patterns emerge. First, subgoal-planning-based methods (Hierarchical FBC and SHARSA) consistently achieve higher success rates than the latent-action-based approach (GCLAOM) across most environments. Second, GCLAOM exhibits greater robustness in action-limited settings, as its success rate remains relatively stable across the different data ratios, suggesting that latent-action methods are less sensitive to the proportion of labeled data. Third, methods that employ value-based representations, such as HIQL, achieve the highest success rates when a larger fraction of action labels is available (e.g., 10% and 100% action-labeled), but suf-

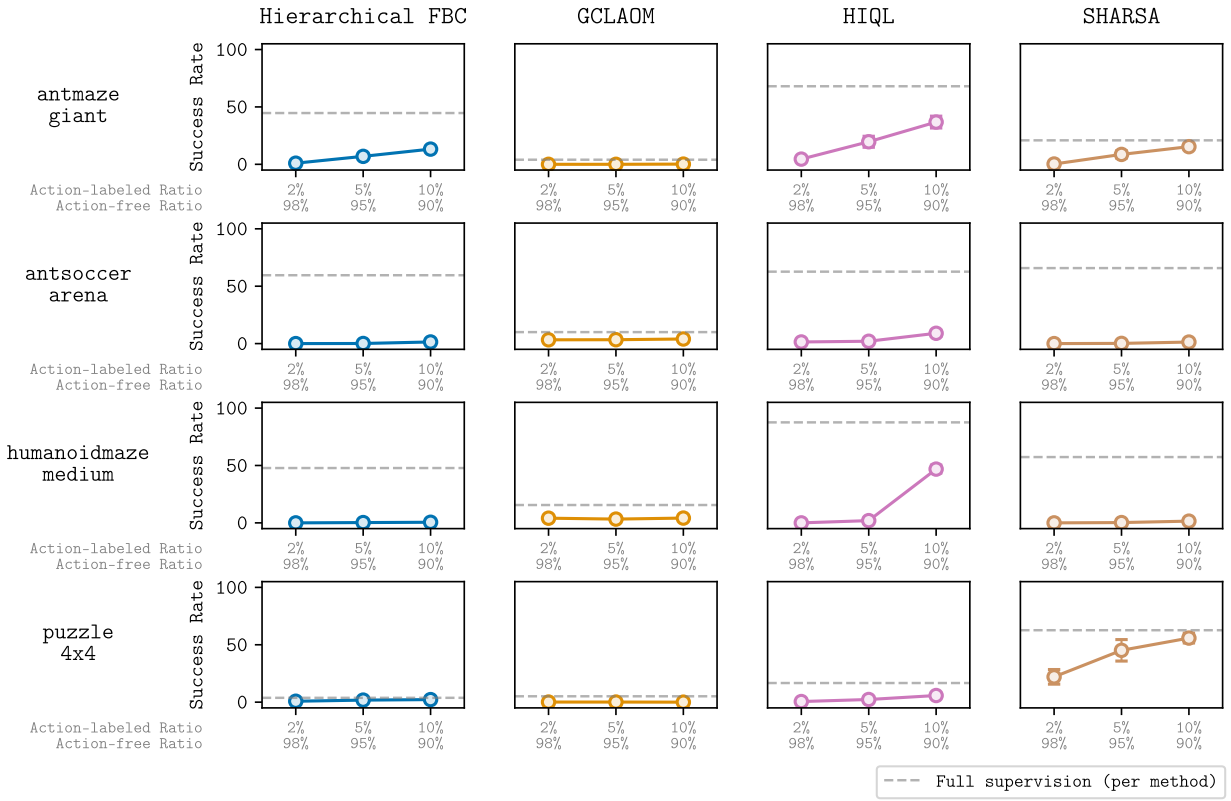


Figure 1: **Existing action-free pretraining methods fail to match full-supervision performance under action-limited settings.** We pretrain four methods using action-free ratios of 98%, 95%, and 90% across four locomotion and manipulation tasks, and fine-tune on the remaining action-labeled data. The dashed line indicates each method's performance under full supervision (100% action-labeled). In all cases, performance under action-limited settings falls significantly short of the full-supervision baseline.

fer a significant performance drop in more restrictive settings and in manipulation tasks such as `puzzle-4x4`, indicating that its effectiveness is more sensitive to the availability of action-labeled transitions.

This analysis suggests that these methods are bottlenecked by the performance of their low-level policy, which in our controlled experiments, has only scarce action-labeled data available for training. While it would be possible to further sweep the action-labeled to action-free ratio and identify the threshold at which a given method matches its fully supervised performance, this falls outside the scope of the present work. Instead, we focus on investigating how to best leverage these techniques to maximize the utility of action-free data.

5.2 Empirical analysis 2: Why do latent actions underperform other methods?

To better understand the behavior of latent actions in action-limited settings in goal-conditioned problems, we compare the action quality produced by the low-level policies of HIQL and GCLAOM to determine whether the capacity to extract action-relevant features from action-free data is a factor in enhancing performance. Notably, both methods share the same low-level behavioral cloning loss, ensuring equivalent learning complexity. We quantify action quality using the mean squared error (MSE) between the predicted and ground-truth actions.

Figure 2 shows that the low-level policy of GCLAOM consistently produces actions of significantly higher quality than that of HIQL across all evaluated environments. Interestingly, this does not translate into higher success rates, as GCLAOM achieves considerably lower success rates than the remaining methods (as

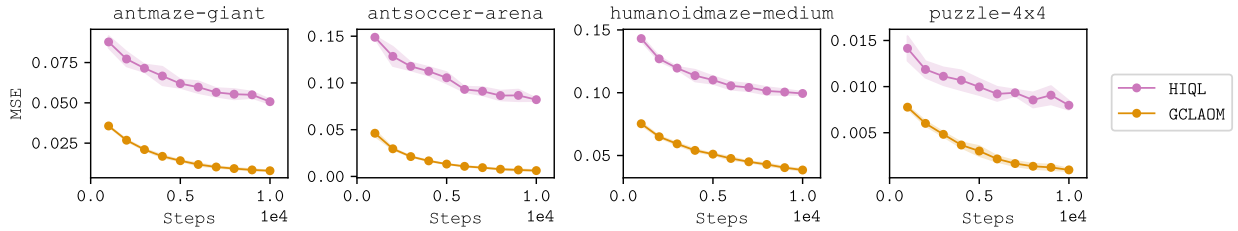


Figure 2: **Latent action learning achieves substantially higher action quality than value-based representations despite lower task success rates.** We measure the mean squared error (MSE↓) of the low-level policy on four goal-reaching tasks to evaluate the capacity of each method to extract action-relevant features from action-free data. GCLAOM consistently achieves lower MSE than HIQL across all environments, yet this does not translate into higher success rates (see Figure 1).

discussed in Section 5.1). This apparent contradiction suggests that superior low-level action prediction is not the primary bottleneck for goal-reaching performance. Rather, it points to the critical role of effective policy horizon reduction: the high-level planners employed by Hierarchical FBC and SHARSA, as well as the value-based subgoal representation of HIQL, are more effective at decomposing the long-horizon goal-reaching problem into manageable subgoals, which ultimately drives higher performance. In contrast, while latent actions provide a better initialization for the low-level policy and yield more accurate action predictions, they do not effectively reduce the policy horizon in a manner comparable to the other approaches, which constitutes their primary limitation in goal-conditioned settings.

Unlike SHARSA and HIQL, which leverage value functions for policy extraction, GCLAOM relies solely on behavioral cloning at the low level. This raises a natural question: could incorporating a value function into GCLAOM allow it to better exploit the high-quality action initialization that its latent action representation provides?

Figure 3 presents results for this variant, which we refer to as GCLAOM with HIQL-PE (policy extraction) (see Appendix A for details), across our action-limited settings. While this variant does not yield a meaningful improvement in success rate over the standard GCLAOM. This result further reinforces our previous finding that latent actions are not effective at reducing the policy horizon, regardless of the policy extraction mechanism employed. The findings from Section 5.1 and Section 5.2 jointly motivate the need for methods that can harness the superior action quality of latent action representations for effective pretraining from action-free data, while also incorporating mechanisms for meaningful policy horizon reduction.

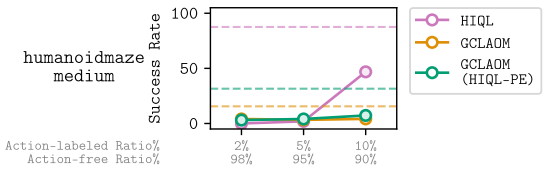


Figure 3: Value-based policy extraction does not recover the performance gap of latent action methods.

5.3 Conditions for Effectively Learning from Passive Data

The preceding discussion yields three key conclusions regarding the design of methods that leverage passive, action-free data for pretraining:

- **Policy horizon reduction is best achieved through high-level planners or value-based subgoal representations.** Methods such as Hierarchical FBC and SHARSA, which predict a future state as a subgoal, and HIQL, which employs a value-based subgoal representation, consistently achieve higher goal-reaching performance. This indicates that explicit policy horizon reduction is a critical factor for success in action-limited settings.

- **Latent action models are effective for low-level action quality improvement, but not for planning.** While latent action representations substantially improve the action quality of the low-level policy, using the latent action model directly for high-level planning yields suboptimal solutions, as it does not effectively reduce the policy horizon. Therefore, the role of the latent action model should be restricted to improving low-level action reconstruction, rather than serving as a planning module.
- **Action-free value learning consistently improves performance.** Method variants that incorporate value functions trained on action-free data achieve higher success rates than their counterparts without value learning, for example, SHARSA outperforms Hierarchical FBC, and GCLAOM (HIQL-PE) outperforms standard GCLAOM. This suggests that value learning from action-free data is an essential component for effective pretraining.

Based on these conclusions, we propose to formalize this intuition as follows. An effective strategy for leveraging passive data is to reduce the policy horizon using a high-level policy, as in SHARSA or HIQL, and pass the resulting subgoal to a latent action model (LAM). In this architecture, the LAM is responsible solely for mapping a (state, subgoal) pair to a latent action, which the low-level policy then decodes into a primitive action. The LAM thus acts as an intermediary module between the high- and low-level policies, improving the quality of low-level actions while the high-level policy retains full responsibility for goal-oriented planning. This decomposition aims to improve goal-reaching performance in action-limited settings using task-agnostic, action-free data.

6 Data-efficient action-free pretraining

Building on the conditions identified in Section 5.3, we propose a method that combines the complementary strengths of the approaches studied in our empirical analysis to enable effective learning from action-free data. In Section 6.2, we evaluate our method across 14 environments under action-limited settings, and in Section 6.3, we compare its goal-reaching performance against the fully action-labeled offline dataset as an upper-bound reference.

6.1 QSIA: a data-efficient method for action-free pretraining

We propose Q-Subgoal, Intent, Action (QSIA), a hierarchical method designed to efficiently leverage action-free data for continuous control. Rather than introducing an entirely new technique, QSIA is grounded in the empirical findings of our analysis: we demonstrate that strategically combining the complementary strengths of existing approaches yields substantial improvements in data efficiency under action-limited settings.

For high-level planning, we adopt the scalable approach proposed in SHARSA, in which a goal-conditioned (GC) value function Q is trained and used to apply rejection sampling over the subgoals proposed by the high-level flow policy, selecting the highest-value subgoal at each step. Specifically, the Q function is trained following the Equation (17), rejection sampling is applied as in Equation (3), and we train the GC value function using IQL as follows:

$$\mathcal{L}^V(V) = \mathbb{E}\Big[L_2^\tau(\mathrm{BCE}(V(s_t, g), \bar{Q}(s_t, s_{t+n}, g)))\Big], \tag{7}$$

where BCE denotes the binary cross-entropy loss, $\bar{Q}$ the target Q network, and L_2^τ denotes the expectile loss with a parameter $\tau \in [0.5, 1) : L_2^\tau(x) = |\tau - 1(x < 0)|x^2$, which can be understood as an asymmetric square loss that penalizes more positive values than negative ones Kostrikov et al. (2022). Next, we adopt an approach similar to LAOM to learn latent actions. Rather than using a shared state representation ψ to encode both s_t and the subgoal s_{t+k} independently as inputs to the IDM, we instead encode the subgoal via a concatenated representation network $\phi([s_t, s_{t+k}])$, which jointly encodes the current state and subgoal pair (see ablation in Appendix C). We therefore learn the latent action as follows:

$$z = \mathrm{IDM}(\psi(s_t), \phi([s_t, s_{t+n}])), \quad \mathcal{L}_{\mathrm{LAL}} = \mathbb{E}_{(s_t, s_{t+1}, s_{t+n}) \sim D}\Big[\parallel \mathrm{FDM}(sg(\psi(s_t)), z) - \psi(s_{t+1}) \parallel^2\Big], \tag{8}$$

Algorithm 1: Q-Subgoal, Intent, Action (QSIA)

Input: Action-free dataset $\mathcal{D}_{\text{passive}}$ or action-labeled dataset $\mathcal{D}$
Initialize forward dynamics model FDM, inverse dynamics model IDM with built-in state representation ψ and ϕ, value functions Q and V, high-policy $\pi^h(s_{t+n}|s, g)$ and low-level policy $\pi^\ell(a_t|z_t)$

while not converged **do**
  Sample batch $\{(s_t, s_{t+1}, \ldots, s_{t+n}, g)\}$ from $\mathcal{D}_{\text{passive}}$;
  if action-labeled dataset $\mathcal{D}$ is available **then**
    Sample batch $\{(s_t, a_t, s_{t+1}, \ldots, s_{t+n}, g)\}$ from $\mathcal{D}$;
  /* Value learning */
  Update $Q(s, s_{t+n}, g)$ by minimizing Equation (17);
  Update $V(s_t, g)$ by minimizing Equation (16);
  /* Latent action learning */
  Update IDM and FDM with $z_t = \text{IDM}(\psi(s_t), \phi([s_t, s_{t+n}]))$ by minimizing Equation (8);
  /* Hierarchical flow behavioral cloning */
  Update $\pi^h(s_{t+n} \mid s_t, g)$ with flow behavioral cloning loss (Equation (19));
  Update $\pi^\ell(a_t \mid z_t)$ with flow behavioral cloning loss (Equation (20)) **only if** $\mathcal{D}$ is available;

where ψ denotes the state representation, ϕ defines the subgoal representation, and sg denotes the stop-gradient function. Finally, we train a hierarchical policy with flow behavioral cloning. We provide the pseudocode for QSIA in Algorithm 1.

6.2 Result on action-limited settings

We evaluate the methods for learning from action-free data described in Section 4 across 14 goal-reaching benchmark tasks under the three action-limited settings introduced in Section 5.1. Table 1 reports success rates across varying proportions of action-labeled and action-free data. Overall, these results demonstrate that strategically combining complementary techniques for action-free learning substantially improves data efficiency in action-limited settings.

Among subgoal planning methods, SHARSA consistently outperforms HGCFBC across most environments, confirming that incorporating a value function for rejection sampling improves performance, particularly on manipulation tasks. A similar trend is observed for latent action methods: GCLAOM (HIQL-PE) generally outperforms standard GCLAOM, further reinforcing that goal-conditioned value learning improves success rates in action-limited settings regardless of the underlying action representation.

Our method, QSIA, which integrates subgoal planning with latent action representations, achieves the highest success rate on the majority of tasks across all three action-limited settings, while maintaining competitive performance even at the most restrictive 2% action-labeled ratio. These results demonstrate that a strong high-level planner enables the low-level policy to effectively exploit the initialization provided by the pre-trained latent action model, thereby improving goal-reaching performance under action-limited settings. Notably, QSIA is the only method to achieve non-zero success rates across nearly all evaluated tasks, with the sole exception of `humanoidmaze-giant`.

6.3 Action-limited results comparison with full supervision

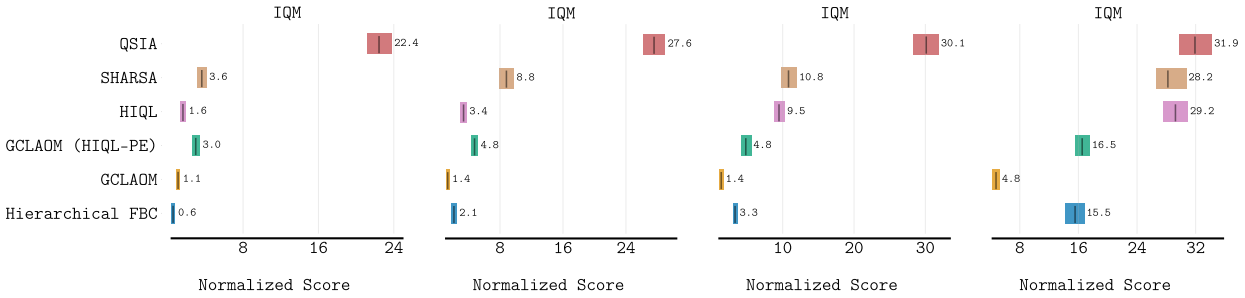
To further characterize the ability of each method to leverage action-free data, we evaluate all methods on the original OGBench offline datasets without any action-label partitioning, and use these results as a full-supervision reference for computing aggregate performance metrics.

Figure 4 reports the interquartile mean (IQM) (Agarwal et al., 2021) for each method across the three action-limited settings and the full-supervision setting. Under full supervision, HIQL, SHARSA, and QSIA achieve comparable performance, confirming that all three methods are competitive when action labels are abundant.

Table 1: **QSIA consistently outperforms all baselines across diverse goal-reaching tasks and action-limited settings.** We report success rates (↑) for six methods across diverse tasks. QSIA achieves the highest success rate across most tasks in all three settings. Numbers at or above 95% of the best value in the row are highlighted in the method's color. The results are averaged over 5 different seeds and we show the standard deviations after the ± sign.

Dataset	**Action-limited**	HGCFBC	GCLAOM	GCLAOM (HIQL-PE)	HIQL	SHARSA	QSIA (ours)
antmaze-large	2% / 98%	27 ±7	7 ±1	26 ±3	50 ±2	21 ±5	52 ±3
	5% / 95%	50 ±3	11 ±3	29 ±2	69 ±8	31 ±2	39 ±2
	10% / 90%	56 ±2	10 ±2	24 ±4	76 ±6	34 ±4	44 ±6
antmaze-giant	2% / 98%	0 ±0	0 ±0	0 ±0	4 ±1	0 ±0	12 ±4
	5% / 95%	6 ±1	0 ±0	0 ±0	19 ±4	8 ±3	19 ±7
	10% / 90%	13 ±1	0 ±0	0 ±0	36 ±5	15 ±2	20 ±3
antsoccer-arena	2% / 98%	0 ±0	3 ±0	14 ±2	1 ±0	0 ±0	46 ±4
	5% / 95%	0 ±0	3 ±0	19 ±2	2 ±1	0 ±0	55 ±3
	10% / 90%	1 ±0	3 ±1	26 ±5	8 ±2	1 ±0	58 ±3
antsoccer-medium	2% / 98%	0 ±0	0 ±0	2 ±1	0 ±0	0 ±0	7 ±1
	5% / 95%	0 ±0	0 ±0	3 ±0	0 ±0	0 ±0	9 ±2
	10% / 90%	0 ±0	0 ±0	1 ±1	2 ±0	0 ±0	11 ±1
humanoidmaze-medium	2% / 98%	0 ±0	4 ±1	3 ±1	0 ±0	0 ±0	50 ±4
	5% / 95%	0 ±0	3 ±1	4 ±0	1 ±0	0 ±0	53 ±4
	10% / 90%	0 ±0	4 ±1	7 ±1	46 ±3	1 ±0	59 ±0
humanoidmaze-large	2% / 98%	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	7 ±2
	5% / 95%	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	9 ±3
	10% / 90%	0 ±0	0 ±0	0 ±0	5 ±2	0 ±0	13 ±2
humanoidmaze-giant	2% / 98%	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0
	5% / 95%	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0	0 ±0
	10% / 90%	0 ±0	0 ±0	0 ±0	4 ±1	0 ±0	0 ±0
cube-single	2% / 98%	15 ±4	7 ±0	6 ±1	5 ±1	21 ±5	20 ±2
	5% / 95%	14 ±3	6 ±2	10 ±2	7 ±2	36 ±6	36 ±2
	10% / 90%	19 ±3	7 ±2	9 ±1	11 ±1	54 ±3	56 ±4
cube-double	2% / 98%	0 ±0	0 ±0	2 ±0	0 ±0	4 ±1	9 ±2
	5% / 95%	2 ±1	1 ±0	8 ±2	0 ±0	9 ±2	16 ±1
	10% / 90%	3 ±1	1 ±0	10 ±2	2 ±1	9 ±1	14 ±1
cube-triple	2% / 98%	0 ±0	1 ±0	8 ±3	0 ±0	0 ±0	3 ±1
	5% / 95%	0 ±0	1 ±1	10 ±0	1 ±0	2 ±1	3 ±1
	10% / 90%	1 ±0	0 ±1	8 ±5	1 ±0	1 ±0	3 ±2
scene	2% / 98%	10 ±2	3 ±1	18 ±2	9 ±2	6 ±1	22 ±4
	5% / 95%	16 ±4	6 ±1	24 ±10	19 ±3	13 ±4	31 ±6
	10% / 90%	23 ±5	7 ±3	48 ±6	24 ±5	16 ±6	27 ±7
puzzle-3x3	2% / 98%	4 ±1	0 ±0	2 ±0	8 ±1	57 ±9	71 ±5
	5% / 95%	6 ±1	0 ±0	2 ±0	11 ±1	66 ±7	65 ±7
	10% / 90%	7 ±0	0 ±0	1 ±1	16 ±2	55 ±11	59 ±17
puzzle-4x4	2% / 98%	0 ±0	0 ±0	0 ±0	0 ±0	22 ±6	54 ±9
	5% / 95%	1 ±1	0 ±0	0 ±0	2 ±1	45 ±9	66 ±4
	10% / 90%	2 ±1	0 ±0	0 ±0	5 ±1	55 ±4	72 ±3
puzzle-4x5	2% / 98%	0 ±0	0 ±0	0 ±0	1 ±0	5 ±2	17 ±0
	5% / 95%	0 ±0	0 ±0	0 ±0	1 ±0	11 ±4	16 ±2
	10% / 90%	0 ±0	0 ±0	0 ±0	2 ±1	14 ±3	18 ±0

However, a clear performance gap emerges under action-limited settings: while HIQL and SHARSA degrade substantially as the proportion of labeled data decreases, QSIA maintains significantly higher IQM scores



(a) 2%/98% (b) 5%/95% (c) 10%/90% (d) Full supervision

Figure 4: **Performance comparison between action-limited and full supervision settings.** Interquartile mean (IQM) (Agarwal et al., 2021) of normalized scores aggregated over all 14 tasks, shown for three action-limited settings (a–c) and the full-supervision baseline (d).

Table 2: QSIA retains the largest fraction of full-supervision performance across all action-limited settings.

Method	2% / 98%	5% / 95%	10% / 90%
QSIA (ours)	70%	86%	94%
SHARSA	12%	31%	38%
HIQL	5%	11%	32%
GCLAOM (HIQL-PE)	18%	28%	29%
GCLAOM	23%	28%	29%
Hierarchical FBC	3%	13%	21%

across all three ratios. This suggests that combining a strong high-level planner with a pretrained latent action model substantially improves the agent's ability to exploit task-agnostic, action-free data.

To quantify this degradation more precisely, we define the *retention rate* as:

$$RR = \frac{S_{AL}}{S_F}, \tag{9}$$

where S_{AL} is the IQM score under the action-limited setting and S_F is the IQM score under full supervision. Table 2 reports the retention rate for each method across all three settings. QSIA achieves the highest retention in all three cases, reaching 94% retention with only 10% action labels, an improvement of approximately 56% over the best competing method (SHARSA, 38%) at the same ratio. Even at the most restrictive 2% setting, QSIA retains 70% of its full-supervision performance, compared to 23% for the next best method (GCLAOM).

7 Conclusion, future directions, and limitations

In this work, we empirically demonstrate that existing techniques for learning from passive data, such as subgoal planning and representation learning, do not generalize well in isolation when only limited task-agnostic, action-labeled data is available. Our analysis motivates an empirical recipe for more effectively exploiting action-free and action-limited datasets: (a) train a subgoal planner with a value-based policy extraction mechanism (e.g., rejection sampling); (b) incorporate a representation learning method that improves the quality of the low-level policy (e.g., latent action learning); and (c) fine-tune a low-level policy using the available action-labeled data. Following this recipe, we propose QSIA, which integrates these design choices into a unified hierarchical framework and demonstrates significant performance gains over all evaluated baselines. Furthermore, our retention rate analysis shows that pretraining on 90% action-free data and fine-tuning with only 10% action-labeled transitions from the same offline dataset can achieve performance comparable to training on the fully action-labeled dataset.

A limitation of our study is that, while QSIA demonstrates substantial improvements in state-based environments, the findings may not directly transfer to visual environments. In pixel-based settings, agents must additionally acquire perceptual and semantic knowledge from raw observations, which introduces representation learning challenges that are absent in state-based control. Nonetheless, this limitation points to a promising direction for future work. Recent methods have shown that high-level planners (Black et al., 2024; Hatch et al., 2025; Intelligence et al., 2026) and latent action models (Ye et al., 2025; Sun et al., 2026; Garrido et al., 2026) can operate in visual environments and leverage large-scale passive video data. Whether QSIA's hierarchical design can be extended to benefit from such large-scale pretraining remains an interesting avenue for future work.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

Chethan Bhateja, Derek Guo, Dibya Ghosh, Anikait Singh, Manan Tomar, Quan Vuong, Yevgen Chebotar, Sergey Levine, and Aviral Kumar. Robotic offline RL from internet videos via value-function pre-training. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023. URL https://openreview.net/forum?id=Rfc9zK6PNO.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. Imitating latent policies from observation. In *International Conference on Machine Learning*, pp. 1755–1763. PMLR, 2019.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. *arXiv preprint arXiv:2601.05230*, 2026.

Dibya Ghosh, Chethan Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 11321–11339, 2023.

Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Conference on Robot Learning*, pp. 1025–1037. PMLR, 2020.

Kyle B Hatch, Ashwin Balakrishna, Oier Mees, Suraj Nair, Seohong Park, Blake Wulfe, Masha Itkina, Benjamin Eysenbach, Sergey Levine, Thomas Kollar, et al. Ghil-glue: Hierarchical control with filtered subgoal images. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9516–9524. IEEE, 2025.

Sukjun Hwang, Brandon Wang, and Albert Gu. Dynamic chunking for end-to-end hierarchical sequence modeling. In *The Fourteenth International Conference on Learning Representations*, 2026.

Physical Intelligence, B Ai, A Amin, R Aniceto, A Balakrishna, G Balke, K Black, G Bokinsky, S Cao, T Charbonnier, et al. π0. 7: a steerable generalist robotic foundation model with emergent capabilities, 2026. *arXiv preprint arXiv:2604.15483*, 2026.

Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, volume 2, pp. 1094–8, 1993.

Hanjung Kim, Lerrel Pinto, and Seon Joo Kim. Hierarchical latent action model. In *ICLR 2026 the 2nd Workshop on World Models: Understanding, Modelling and Scaling*, 2026.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*, pp. 45–73. Springer, 2012.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pp. 5639–5650. PMLR, 2020.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. In *Proceedings of International Conference on Learning Representations*, 2019.

Anthony Liang, Pavel Czempin, Matthew Hong, Yutai Zhou, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pp. 1113–1132. Pmlr, 2020.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*, 2023.

Ajay Mandlekar, Fabio Ramos, Byron Boots, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Dieter Fox. Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4414–4420. IEEE, 2020.

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. *arXiv preprint arXiv:2406.17098*, 2024.

Vivek Myers, Bill Zheng, Benjamin Eysenbach, and Sergey Levine. Offline goal-conditioned reinforcement learning with quasimetric representations. *Advances in Neural Information Processing Systems*, 38:19654–19679, 2026.

Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Lyubaykin Nikita, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. In *Forty-second International Conference on Machine Learning*, 2025.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline goal-conditioned rl with latent states as actions. *Advances in Neural Information Processing Systems*, 36:34866–34891, 2023.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Seohong Park, Aditya Oberai, Pranav Atreya, and Sergey Levine. Transitive rl: Value learning via divide and conquer. *arXiv preprint arXiv:2510.22512*, 2025b.

Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes rl scalable. *Advances in Neural Information Processing Systems*, 38:8350–8389, 2026.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Tom Schaul, Dan Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pp. 1312–1320, 2015.

Jurgen Schmidhuber. Learning to generate sub-goals for action sequences. In *Artificial neural networks*, pp. 967–972, 1991.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *The Twelfth International Conference on Learning Representations*, 2024.

Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. In *International Conference on Machine Learning*, pp. 19561–19579. PMLR, 2022.

Martin Stolle and Doina Precup. Learning options in reinforcement learning. In *International Symposium on abstraction, reformulation, and approximation*, pp. 212–223. Springer, 2002.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

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*, pp. 36411–36430, 2023.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, volume 2025, pp. 28213–28239, 2025.

A Methods

We describe the offline goal-conditioned RL (GCRL) algorithms for action-free pretraining considered in this work. We train these methods in two stages: first, we pretrain the components that do not require action-labeled data using action-free data; then, we fine-tune all components with an action-limited dataset.

HGCFBC, SHARSA (Park et al., 2026)**, and HIQL** (Park et al., 2023). We employ the original implementations provided by Park et al. (2025a; 2026) without modification, and refer the reader to those works for full technical details.

GCLAOM. We adapt LAOM (Nikulin et al., 2025), a latent action learning method that extends LAPO (Schmidt & Jiang, 2024) with several modifications to improve robustness to distractors and performance on continuous control tasks. LAOM consists of four main components: an inverse dynamics model (IDM), a forward dynamics model (FDM) operating in latent space, a state representation ψ, and a latent action policy $\pi^h_{\text{LAL}}(z_t \mid s_t)$. Key modifications over LAPO include a multi-step IDM, a larger latent action dimension, data augmentation, and supervision of the IDM via a small action-labeled dataset through an auxiliary linear layer. Notably, Nikulin et al. (2025) find that the latent action dimensionality does not significantly degrade performance when action-labeled supervision is available; we therefore fix the dimensionality to 128 across all experiments.

Since LAOM was originally proposed for standard offline RL (Levine et al., 2020), we extend it to the offline GCRL setting by conditioning the latent action policy on the goal: $\pi^h_{\text{LAL}}(z_t \mid s_t, g)$. The training objectives are as follows:

$$z_t = \text{IDM}(\psi(s_t), \psi(s_{t+n})), \quad \mathcal{L}_{\text{LAL}} = \mathbb{E}_{(s_t, s_{t+1}, s_{t+n}) \sim \mathcal{D}}\Big[\parallel \text{FDM}(\text{sg}(\psi(s_t)), z) - \bar{\psi}(s_{t+1}) \parallel^2 \Big], \tag{10}$$

$$J_{\pi^h_{\text{LAL}}} = \mathbb{E}_{(s_t, s_{t+n}) \sim \mathcal{D}, g \sim p^{\mathcal{D}}(g|\tau)}[\log(\pi^h(z_t|s_t, g))], \tag{11}$$

$$J_{\pi^\ell_{\text{LAL}}} = \mathbb{E}_{(s_t, a_t, s_{t+n}) \sim \mathcal{D}}[\log(\pi^\ell(a_t|z_t))], \tag{12}$$

where s_{t+n} denotes a state n steps ahead of s_t, ψ is the learned state representation, $\bar{\psi}$ is its target network, $\text{sg}(\cdot)$ denotes the stop-gradient operator, and the low-level policy is used to supervise the IDM on the action-labeled subset.

GCLAOM (HIQL-PE). This variant replaces GCLAOM's behavioral cloning policy extraction with HIQL-style policy extraction. Specifically, it trains a goal-conditioned value function using implicit value learning (IVL) (Park et al., 2025a) and extracts hierarchical policies via advantage-weighted regression (AWR) (Peng et al., 2019):

$$\mathcal{L}_{IVL}(\theta_V) = \mathbb{E}_{(s_t, s_{t+1}) \sim \mathcal{D}, g \sim p^{\mathcal{D}}(g|\tau)}[L_2^\tau(r(s_t, g) + \gamma V_{\bar{\theta}_V}(s_{t+1}, g) - V_{\theta_V}(s_t, g))], \tag{13}$$

$$J_{\pi^h_{\text{LAL}}} = \mathbb{E}_{(s_t, s_{t+n}) \sim \mathcal{D}, g \sim p^{\mathcal{D}}(g|\tau)}[\exp(\beta(V(s_{t+n}, g) - V(s_t, g)))\log(\pi^h(z_t|s_t, g))], \tag{14}$$

$$J_{\pi^\ell_{\text{LAL}}} = \mathbb{E}_{(s_t, a_t, s_{t+n}) \sim \mathcal{D}}[\exp(\beta(V(s_{t+1}, s_{t+n}) - V(s_t, s_{t+n})))\log(\pi^\ell(a_t|z_t))], \tag{15}$$

where L_2^τ denotes the expectile loss (Kostrikov et al., 2022), β is the AWR inverse temperature hyperparameter, and z_t is the latent action produced by the IDM as defined in Equation (10). All other components remain identical to GCLAOM.

QSIA. Our proposed method, QSIA, integrates subgoal planning from SHARSA (Park et al., 2026) with latent action learning, modeling a hierarchical policy using flow-based behavioral cloning. We describe each training component individually.

Value functions. We train goal-conditioned value functions using IQL-style losses:

$$\mathcal{L}^V(V) = \mathbb{E}_{\substack{(s_t, \ldots, s_{t+n}) \sim \mathcal{D}, \\ g \sim p^{\mathcal{D}}(g|s_t, s_{t+n})}}\Big[L_2^\tau(\text{BCE}(V(s_t, g), \bar{Q}(s_t, s_{t+n}, g)))\Big], \tag{16}$$

$$\mathcal{L}^Q(Q) = \mathbb{E}_{\substack{(s_t,\ldots,s_{t+n})\sim\mathcal{D},\\ g\sim p^{\mathcal{D}}(g|s_t,s_{t+n})}}\Big[\mathrm{BCE}(Q(s_t,s_{t+n},g), \sum_{i=0}^{n-1}\gamma^i r(s_{t+i},g) + \gamma^n V(s_{t+n},g))\Big]. \tag{17}$$

Latent action learning. We train the IDM and FDM with the following loss:

$$z_t^{\mathrm{LAL}} = \mathrm{IDM}(\psi(s_t), \phi([s_t, s_{t+n}])), \quad \mathcal{L}_{\mathrm{LAL}} = \mathbb{E}_{(s_t,s_{t+1},s_{t+n})\sim\mathcal{D}}\Big[\parallel \mathrm{FDM}(\mathrm{sg}(\psi(s_t)), z_t^{\mathrm{LAL}}) - \bar{\psi}(s_{t+1}) \parallel^2\Big]. \tag{18}$$

Hierarchical policy. Following HFBC (Park et al., 2026), we model the high-level and low-level policies as flow-based vector fields, $v^h(t,s,z,g)$ and $v^\ell(t,z^{\mathrm{LAL}},z)$, trained with the following flow behavioral cloning losses:

$$J^h = \mathbb{E}_{\substack{(s_t,\ldots,s_{t+n})\sim\mathcal{D}, g\sim p^{\mathcal{D}}(g|s_t,s_{t+n}),\\ z\sim\mathcal{N}(0,I_m), t\sim\mathrm{Unif}([0,1]),\\ w^t=(1-t)z+ts_{t+n}}}\Big[\parallel v^h(t,s_h,w^t,g) - (s_{t+n}-z)\parallel_2^2\Big], \tag{19}$$

$$J^\ell = \mathbb{E}_{\substack{(s_t,a_t,\ldots,s_{t+n})\sim\mathcal{D}, z\sim\mathcal{N}(0,I_d),\\ t\sim\mathrm{Unif}([0,1]), a^t=(1-t)z+ta_t}}\Big[\parallel v^\ell(t,z_t^{\mathrm{LAL}},a^t) - (a_t - z)\parallel_2^2\Big], \tag{20}$$

where $\mathrm{Unif}([0,1])$ defines the uniform distribution in the interval $[0,1]$, and z_t^{LAL} is the latent action of Equation (18). Note that the low-level policy is conditioned on the latent action z_t^{LAL} rather than a (state, subgoal) pair as it is formulated in HGCFBC and SHARSA.

Test-time subgoal selection. Following SHARSA, we apply rejection sampling at test time to select the highest-value subgoal proposed by the high-level policy:

$$\pi^h(s,g) \stackrel{d}{=} \underset{w_1,\cdots,w_N:\, w_i\sim\pi_\beta^h(w|s,g)}{\arg\max} Q(s,w_i,g), \tag{21}$$

where N is the number of candidate subgoal samples, $\stackrel{d}{=}$ denotes equality in distribution, and π_β denotes the goal-conditioned behavioral cloning policy.

B Implementation details

We implement QSIA on top of the reference codebases provided by Park et al. (2025a; 2026) and generally keep the base hyperparameters consistent with those works.

Table 3: **Common hyperparameters.**

Hyperparameter	**Value**
Gradient steps (pre-training)	900000,
Gradient steps (action-limited finetuning)	$10000, 8000, 5000$ (10%, 5%, and 2% action labeled ratio respectively)
Gradient steps (full action-labeled dataset)	1000000
Optimizer	Adam (Kingma, 2014)
Learning rate	0.0003
Minibatch size	1024
Target network update rate	0.005 (value function, state representation ψ)
Expectile κ	0.7
Subgoal step κ	100 (humanoidmaze), 25 (antmaze, antsoccer), 10 (manipulation)
Goal representation dimension (QSIA, HIQL)	128
State representation dimension (QSIA, GCLAOM, GCLAOM (HIQL-PE))	128
Latent action dimension (QSIA, GCLAOM, GCLAOM (HIQL-PE))	128
MLP size	[512, 512, 512]
Policy $(p^{\mathcal{D}}_{\mathrm{cur}}, p^{\mathcal{D}}_{\mathrm{geom}}, p^{\mathcal{D}}_{\mathrm{traj}}, p^{\mathcal{D}}_{\mathrm{rand}})$ ratio (QSIA, SHARSA)	(0.0, 0.5, 0.0, 0.5)
Value $(p^{\mathcal{D}}_{\mathrm{cur}}, p^{\mathcal{D}}_{\mathrm{geom}}, p^{\mathcal{D}}_{\mathrm{traj}}, p^{\mathcal{D}}_{\mathrm{rand}})$ ratio (QSIA, SHARSA)	(0.2, 0.8, 0.0, 0.0)
Policy $(p^{\mathcal{D}}_{\mathrm{cur}}, p^{\mathcal{D}}_{\mathrm{geom}}, p^{\mathcal{D}}_{\mathrm{traj}}, p^{\mathcal{D}}_{\mathrm{rand}})$ ratio (others)	(0.0, 0.0, 0.1, 0.0)
Value $(p^{\mathcal{D}}_{\mathrm{cur}}, p^{\mathcal{D}}_{\mathrm{geom}}, p^{\mathcal{D}}_{\mathrm{traj}}, p^{\mathcal{D}}_{\mathrm{rand}})$ ratio (HIQL, GCLAOM (HIQL-PE))	(0.2, 0.5, 0.0, 0.3)

Training and evaluation. We generally follow the evaluation protocols of Park et al. (2025a). The number of training steps depends on the experimental setting. When training on the full action-labeled dataset, we train for 1M steps and evaluate every 200k steps. Under action-limited settings, we first pretrain on action-free data for 900k steps and then fine-tune for 10k, 8k, and 5k steps for the 10%, 5%, and 2% action-labeled

ratios, respectively. In these settings, we evaluate every 2000 steps, or every 1000 steps when fine-tuning with 2% action labels. For all results, we report the average success rate over the last three evaluations (the last two for the full action-labeled setting). We use 15 episodes for each of the five test-time goals provided by OGBench (Park et al., 2025a). All results are averaged across 5 random seeds (3 seeds for full supervision).

Hyperparameters. We mostly follow the base hyperparameters from OGbench (Park et al., 2025a). The exact hyperparameter values used in this paper, along with the number of gradient steps for the action-limited experiments, are reported in Table 3.

C Ablation studies of QSIA

In this section, we evaluate three ablation studies regarding the design choices of QSIA. All results are evaluated on a full action-labeled setting.

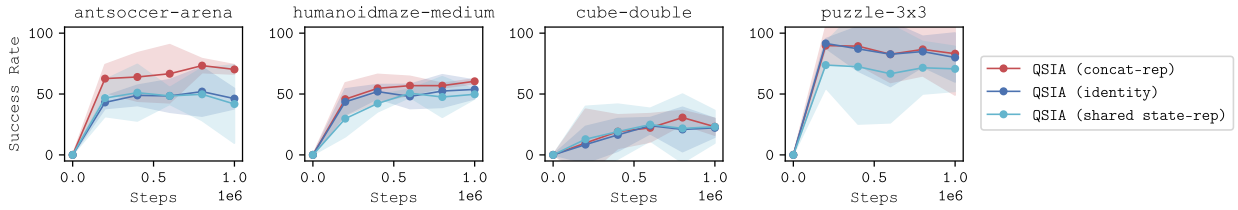


Figure 5: Ablation study on the goal representations.

Subgoal representation. QSIA uses a concatenated subgoal representation, $\phi([state, subgoal])$, to map the state-goal pair into a latent vector, following a design similar to HIQL (Park et al., 2023) rather than using only the state or the subgoal independently. We ablate this design choice in Figure 5, where we compare three types of representations for QSIA: a concatenation representation $\phi([s_t, s_{t+k}])$, a shared state representation $\phi(s_{t+k}) = \psi(s_{t+k})$ (as in LAOM Appendix A), and the raw subgoal s_{t+k} (identity). The results show that the concatenated representation generally improves the performance of our method. As in HIQL (Park et al., 2023), we hypothesize that this choice provides the IDM with more contextual information and, as our empirical results indicate, leads to stronger performance.

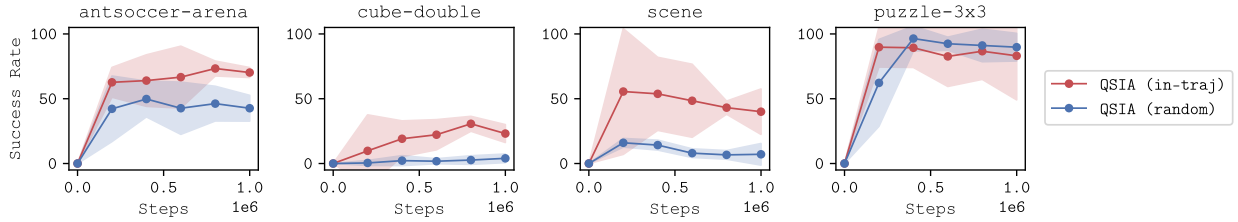


Figure 6: Ablation study on the value goal distribution.

Value goal distribution. While it is common practice to mix a small proportion of randomly sampled states into the goal distribution (Park et al., 2025a; 2026), we find that using only in-trajectory states as goals generally improves performance. The results in Figure 6 suggests that including random states for value training may hurt the quality of the learned value function, and it is consistent with previous findings in (Park et al., 2025b).

Value learning method. In Figure 7, we compare SARSA and IQL as the value learning method used in QSIA. The results show that while SARSA performs comparably on antsoccer-arena and cube-double, offline RL value learning methods such as IQL are crucial for achieving strong performance on environments such as `scene` and `puzzle-3x3`.

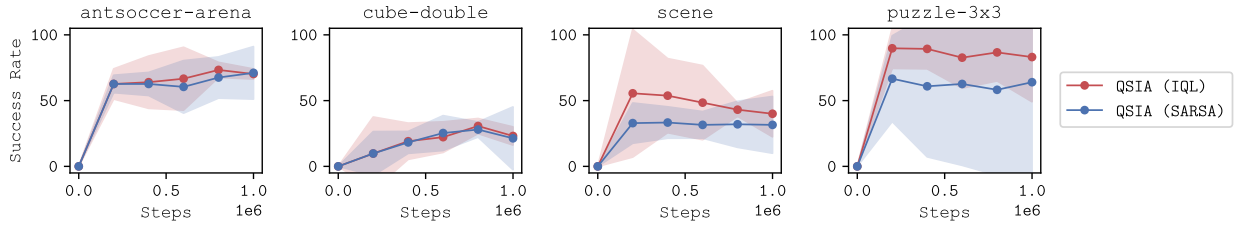


Figure 7: Ablation study on value learning method.

D Additional results.

We report results on the fully action-labeled dataset across a diverse set of goal-reaching tasks from OGBench in Table 4. Figures 8 and 9 present these results graphically, summarizing the data from Tables 4 and 1.

Table 4: **Results on full action-labeled dataset.** We report success rates (↑) for six methods across diverse tasks. Numbers at or above 95% of the best value in the row are highlighted in the method's color.

Dataset	HGCFBC	GCLAOM	GCLAOM (HIQL-PE)	HIQL	SHARSA	QSIA (ours)
antmaze-large	74 ±1	45 ±1	71 ±0	90 ±1	37 ±1	39 ±3
antmaze-giant	44 ±1	3 ±1	1 ±1	68 ±4	20 ±2	22 ±5
antsoccer-arena	59 ±2	9 ±0	42 ±0	62 ±1	65 ±4	71 ±0
antsoccer-medium	7 ±2	3 ±0	5 ±1	12 ±1	9 ±3	12 ±1
humanoidmaze-medium	47 ±1	15 ±1	31 ±2	87 ±1	57 ±2	58 ±0
humanoidmaze-large	6 ±2	1 ±1	3 ±1	40 ±7	11 ±2	13 ±0
humanoidmaze-giant	1 ±1	1 ±0	0 ±0	14 ±2	0 ±0	0 ±0
cube-single	21 ±6	8 ±2	30 ±1	22 ±1	45 ±11	35 ±12
cube-double	4 ±1	1 ±0	21 ±2	5 ±1	17 ±2	26 ±1
cube-triple	1 ±1	0 ±0	11 ±1	3 ±2	1 ±1	3 ±1
scene	34 ±3	12 ±2	59 ±12	49 ±6	30 ±4	41 ±3
puzzle-3x3	7 ±2	3 ±0	8 ±4	20 ±3	64 ±18	84 ±9
puzzle-4x4	3 ±1	5 ±1	21 ±2	16 ±2	62 ±11	79 ±4
puzzle-4x5	0 ±0	1 ±0	2 ±0	2 ±0	18 ±0	18 ±0

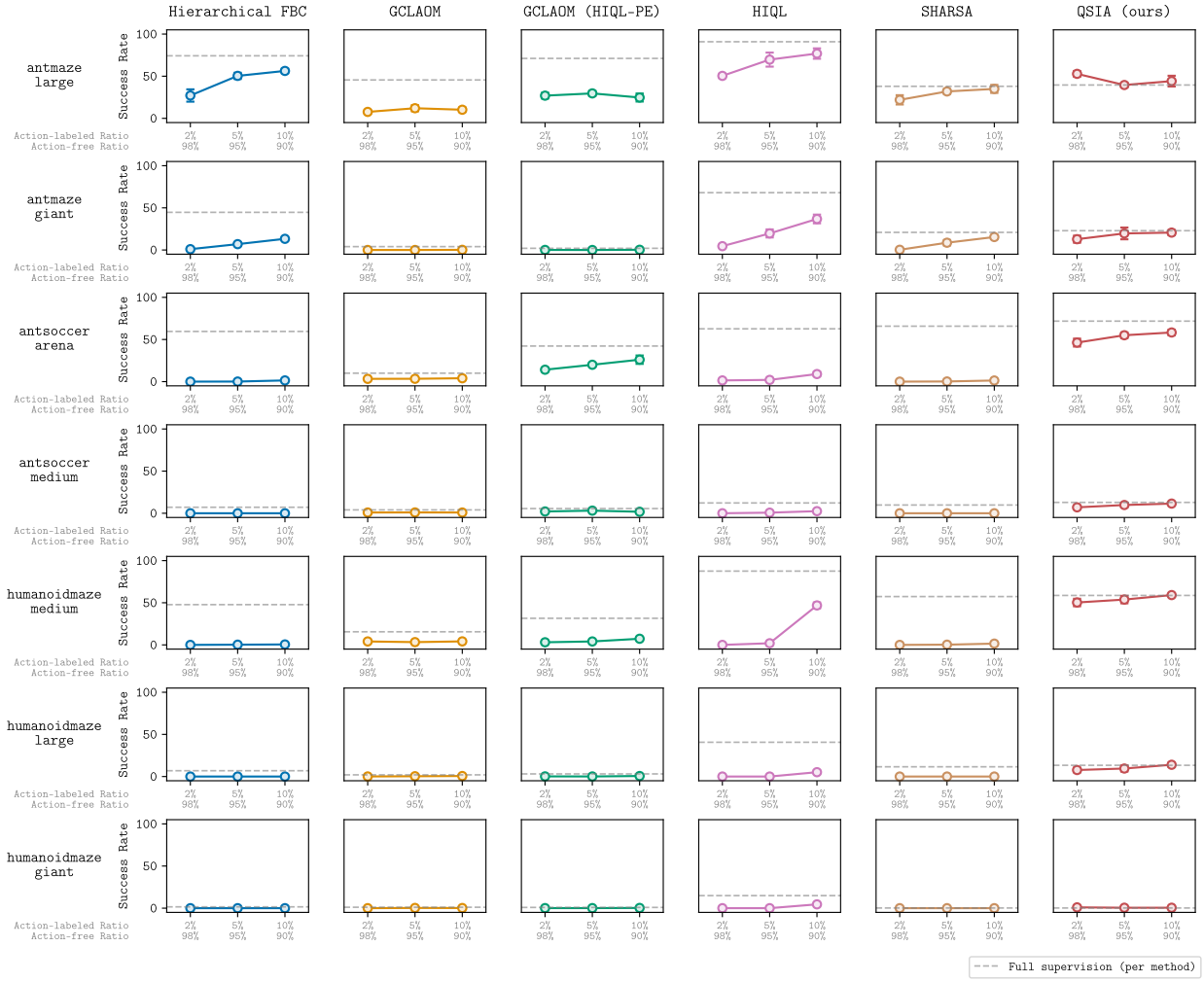


Figure 8: Results on locomotion tasks under action-limited and full action-labeled settings.

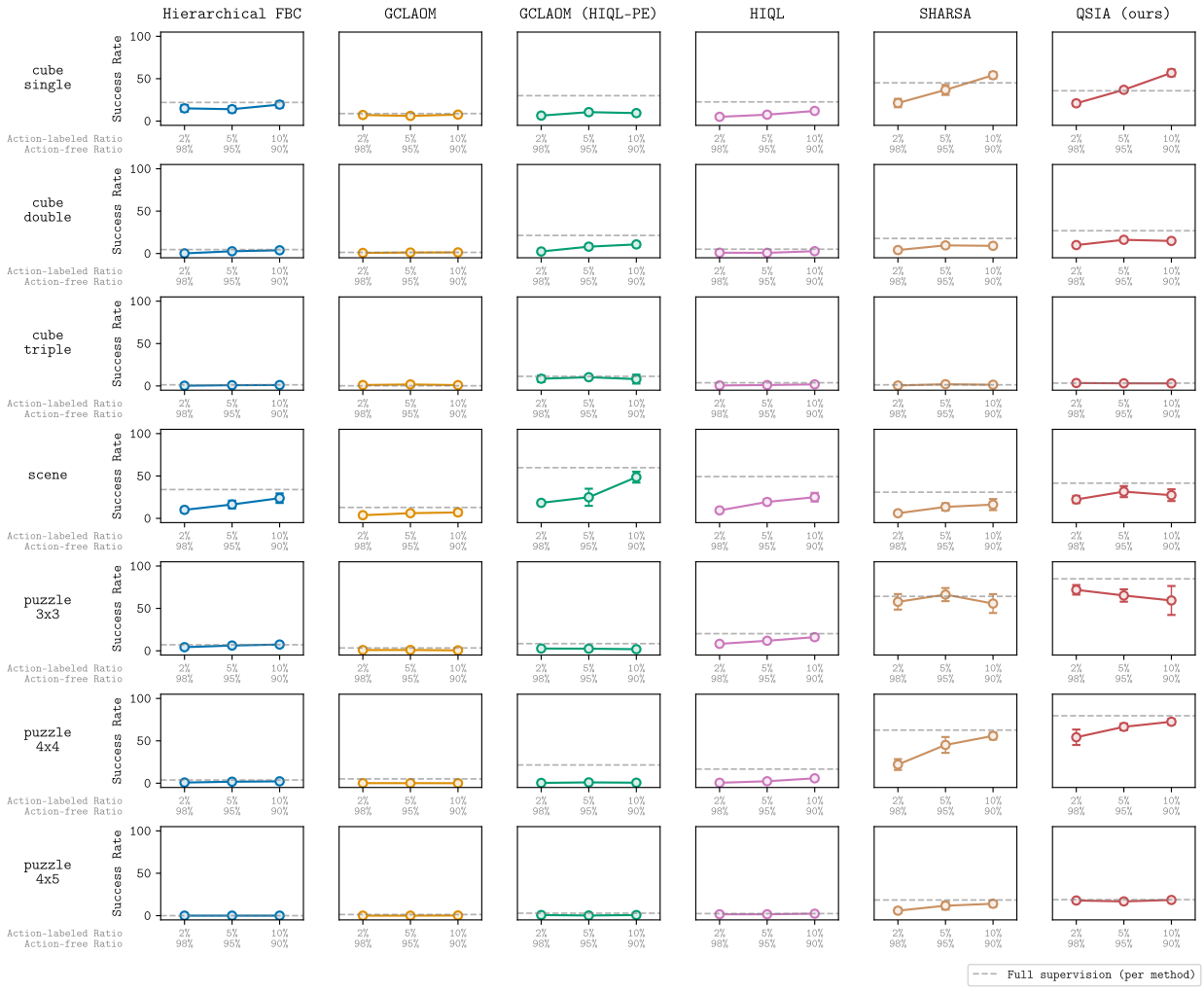


Figure 9: Results on manipulation tasks under action-limited and full action-labeled settings.