DECOUPLING POLICY EXTRACTION FOR OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline RL methods commonly jointly train the actor and critic, where the critic is used to guide the actor toward higher-value actions. This coupled paradigm is well motivated in online RL, where an improved actor collects new data that can further update the critic. However, training data remains fixed in offline RL, making actor-side policy improvement unable to generate new data to correct the critic. Moreover, retaining this coupled paradigm leads to two related challenges. Firstly, actor updates can drift toward out-of-distribution (OOD) actions and amplify critic overestimation. Secondly, conservative value estimation or behavior-cloning regularization creates a difficult trade-off between suppressing OOD actions and selecting high-value actions within the data-supported region. Motivated by this observation, we revisit the conventional offline RL paradigm and propose the **decoupled policy extraction paradigm**, in which the actor is solely trained to model the dataset distribution, while the policy improvement is defined as a test time critic-based reranking operation. Under this paradigm, the actor provides behavior-supported action candidates, while the critic performs value-based selection within this candidate set. Extensive experiments show that the decoupled policy extraction paradigm outperforms both behavior cloning and jointly learned offline RL methods, while remaining effective even with a naive Q-learning critic.

1 INTRODUCTION

Reinforcement learning (RL) learns decision-making policies from reward feedback, which is commonly implemented with an actor–critic architecture (Sutton & Barto, 2018; Li, 2017). The critic estimates the long-term value of actions, while the actor is trained under its guidance to favor better actions. This division of labor is especially natural in online RL, because the critic drives the actor toward high-value actions, while the improved actor in turn expands the critic's data support and corrects its estimation errors through new environment interactions, thereby forming a meaningful closed-loop training process.

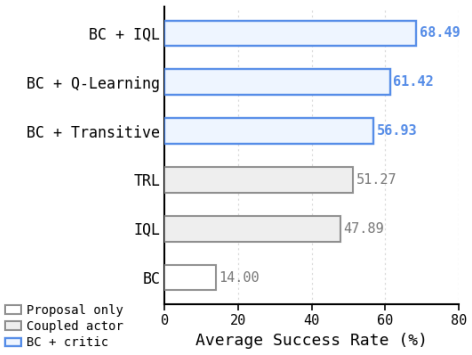


Figure 1: Aggregate performance on 30 offline GCRL tasks. Decoupled policy extraction variants outperform original coupled baselines.

A broad class of offline RL methods has retained such a coupled paradigm (Fujimoto et al., 2019; Kumar et al., 2020; Kostrikov et al., 2022), we believe the rationale for retaining actor-side policy improvement is substantially weaker in offline RL. As the dataset is fixed, improving the actor neither changes the training distribution nor collects new experiences that could correct critic errors. Moreover, retaining this paradigm suffers from two fundamental issues: the **OOD amplification loop** problem and the **support–value trade-off** problem. Firstly, the actor continually shifts toward potentially out-of-distribution actions assigned higher values by the critic, thereby amplifying the effect of critic overestimation and further shifting the policy away from the data distribution, leading to the OOD amplification loop problem. Secondly, existing methods typically introduce conservative value estimation or behavior cloning regularization to control this deviation. However, weak constraints may fail to suppress OOD actions, whereas strong constraints may prevent the policy from selecting high-value actions even within the data-

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.

supported region, resulting in a difficult trade-off between value maximization and distributional support, namely the support–value trade-off problem.

Building on the above observation, we revisit the paradigm design for offline reinforcement learning by decoupling policy improvement from actor learning. Specifically, we propose to learn a behavior cloning actor and learn a critic independently. At inference time, the actor proposes multiple data-supported candidate actions, which are reranked and selected by the critic. In contrast to prior coupled policy improvement methods, we refer to this paradigm as **decoupled policy extraction paradigm**. Under this paradigm, policy improvement is no longer realized through critic-driven actor optimization, but is instead explicitly instantiated as test-time action selection. This decoupling naturally addresses the two issues discussed above. First, because the critic no longer updates the actor, critic overestimation is not recursively amplified through actor optimization, thereby avoiding the OOD amplification loop. Second, distributional support and value maximization are no longer balanced within a single actor objective, avoiding the support–value trade-off inherent in coupled policy improvement.

The aggregate results in Figure 1 demonstrate the advantage of decoupling policy improvement from actor learning. Across 30 offline goal-conditioned tasks, replacing the original critic-guided actor with a decoupled policy extraction variant consistently improves performance. For example, with the IQL critic, the decoupled variant increases the average success rate from 47.89% to 68.49%. Similar improvements with other critic backbones indicate that the benefit is not specific to a particular value-learning objective. It is worth noting that although related inference-time policy optimization approaches have appeared in several existing methods (Hansen-Estruch et al., 2023; Nakamoto et al., 2025), we aim to provide a systematic discussion of RL method mechanisms. The relationship between our paradigm and existing methods will be discussed in Section 6.

Our contributions are as follows:

- We revisit the prevailing coupled policy improvement paradigm in offline RL and identify its limitations. Based on this analysis, we propose a decoupled policy extraction paradigm which suits the offline setting better.
- We instantiate this paradigm with multiple critic backbones and show that decoupled policy extraction consistently improves performance over the corresponding coupled methods across a broad range of offline goal-conditioned tasks.
- The proposed paradigm has important implications for large-scale policies. By delegating policy improvement to a lightweight critic while avoiding repeated updates to the actor, it provides a more computationally efficient approach to offline RL with large policy models.

2 Preliminaries

2.1 Offline Goal-Conditioned Reinforcement Learning

A goal-conditioned reinforcement learning (GCRL) task is commonly modeled as a Markov decision process:

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{G}, r, \gamma, H), \tag{1}$$

where $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{G}$ denote the state, action, and goal spaces; r is the reward function, γ is the discount factor, H is the maximum task horizon. The goal-conditioned policy $\pi_\theta(\cdot \mid s, g)$ is learned to maximize

$$J(\pi) = \mathbb{E}_{g\sim p_G,\, \tau\sim p^\pi(\cdot|g)} \left[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t, g) \right], \qquad T \le H, \tag{2}$$

where $a \in \mathcal{A}$, $s \in \mathcal{S}$ and $g \in \mathcal{G}$. Our paper considers the offline GCRL setting, the policy $\pi_\theta(\cdot|s, g)$ is trained from a fixed dataset $\mathcal{D} \triangleq \{(s_t, a_t, g_t, r_t)_{0\le t\le T}\}$.

2.2 Coupled Policy Extraction

A general form of coupled policy extraction can be abstracted as

$$\max_\theta \quad \mathbb{E}_{a\sim\pi_\theta(\cdot|s,g)} \left[Q_\phi(s, a, g) \right] \quad \text{s.t.} \quad D\left(\pi_\theta(\cdot \mid s, g), \mu_\mathcal{D}(\cdot \mid s, g)\right) \le \epsilon, \tag{3}$$

where Q_ϕ is the critic, $\mu_\mathcal{D}$ denotes the behavior distribution represented by the offline dataset, $D(\cdot, \cdot)$ measures the deviation between the learned policy and the behavior distribution, and ϵ controls the allowable degree of deviation.

This formulation captures the common structure of coupled policy extraction: the actor is optimized toward high-value actions while being constrained to remain close to the offline behavior distribution. In practice, this coupling is commonly realized in two forms: *advantage-weighted regression*, such as IQL (Kostrikov et al., 2022), where value information reweights the imitation of dataset actions; and *BC-regularized value maximization*, as in TD3+BC (Fujimoto & Gu, 2021), where the actor directly maximizes the critic while being regularized toward dataset actions. Concrete formulations are provided in Appendix D.

2.3 Proposal-and-Selection Policies

We define $\pi_\theta(a \mid s, g)$ as a stochastic action proposer based on behavior cloning (BC). Given the current state s_t and goal g, the proposer samples N candidate actions

$$C_N(s_t, g; \pi_\theta) = \{a_t^1, a_t^2, \ldots, a_t^N\}, \qquad a_t^i \sim \pi_\theta(\cdot \mid s_t, g). \tag{4}$$

With an independently learned critic Q_ϕ, the optimal action a^* is selected using the critic-based reranking operation:

$$a_t^* = \arg \max_{a_t^i \in C_N(s_t, g; \pi_\theta)} Q_\phi(s_t, a_t^i, g). \tag{5}$$

This notation makes three factors explicit: the BC-based actor π_θ, the critic ranking function Q, and the candidate budget N. In our experiments, we instantiate π_θ as a frozen behavior-cloned proposer and vary the critic backbone and candidate budget to diagnose offline policy extraction.

3 Diagnosing Coupled Policy Improvement in Offline RL

As discussed in Section 1, jointly trained actor–critic methods are subject to two key limitations: the amplification of critic errors through the actor–critic feedback loop, and the support–value trade-off. In this section, we empirically investigate these two issues separately. We further test whether decoupling proposal learning from critic-based selection changes performance under compute-matched candidate search with a fixed critic. Detailed settings are provided in Appendix G and H.

3.1 Out-of-distribution Amplification Loop

Out-of-distribution (OOD) actions have been recognized as a central challenge in offline reinforcement learning. A key source of this difficulty lies in the interaction between critic estimation error and coupled policy improvement. When the actor assigns probability mass to actions outside the support of the offline dataset, the critic may consequently overestimate some of them. Directly optimizing the actor against such estimates can then increase the probability of these overestimated OOD actions, further shifting the actor toward regions where the critic is less reliable. This creates a self-reinforcing feedback loop between critic error and actor distribution shift. In online RL, such errors can be corrected because the updated policy generates new interactions that provide training data for these newly visited regions. In offline RL, however, the dataset remains fixed, leaving this feedback loop without an equivalent correction mechanism.

We next empirically examine this feedback-loop error amplification in coupled policy improvement methods without explicit pessimistic regularization. Figure 2 compares EDP-BC, EDP-CRR and EDP-TD3, representing BC policy, advantage-weighted regression, and behavior-regularized Q maximization, respectively. Details of EDP-BC, EDP-CRR and EDP-TD3 can be found in Appendix G. Although all methods use the same initialization, after training, the fraction of actions outside the same-checkpoint BC95 region differs; the fraction is 6.8% for EDP-BC, compared with 31.3% for EDP-CRR and 16.8% for EDP-TD3. These results highlight a fundamental vulnerability of coupled policy improvement: directly optimizing the actor against an imperfect offline critic can amplify distribution shift through the actor–critic feedback loop. Moreover, actions beyond BC95 receive higher diagnostic advantages under CRR and higher within-checkpoint Q ranks under TD3. These results provide evidence of critic-aligned actor drift: coupled updates can move the actor beyond the BC reference support, whereas freezing the proposal policy removes this feedback path.

3.2 The Support–Value Trade-off

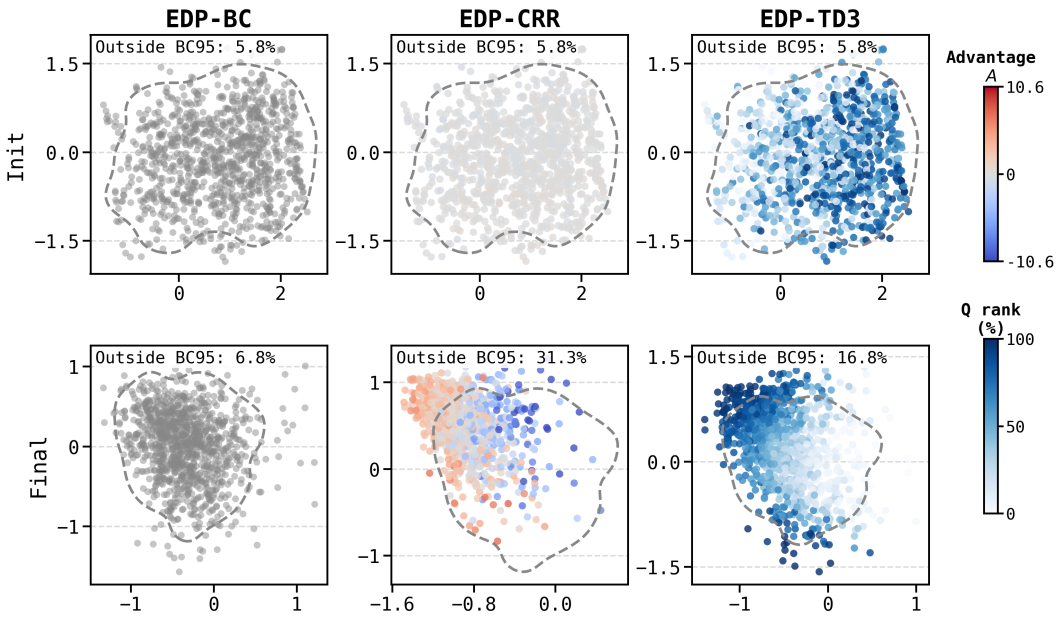


Figure 2: Dashed contours mark the same-checkpoint 95% conformal BC reference regions, with outside rates reported in each panel. Colors encode raw diagnostic advantages for CRR and within-checkpoint $\min(Q_1, Q_2)$ percentile ranks for TD3.

As discussed earlier, under the coupled policy improvement paradigm, out-of-distribution actions can induce a looped error amplification between critic estimation errors and actor updates, thereby leading to actor distribution shift. Existing offline RL methods commonly address this issue through conservative or behavior-cloning regularization. However, we believe that these mechanisms do not eliminate the underlying conflict; instead, they trade off policy improvement against adherence to the support of the offline dataset.

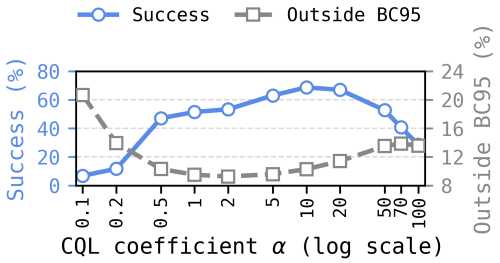


Figure 3: Success and BC support across selected CQL conservatism settings on Cube Double. The coefficient that maximizes success differs from that minimizing Outside BC95, while stronger pessimism eventually reduces success without improving the support diagnostic.

Taking CQL as an example, we examine goal-conditioned CQL (Kumar et al., 2020), where α_{CQL} scales the conservative critic penalty while the SAC (Haarnoja et al., 2018) actor objective remains unchanged. Figure 3 reports task success together with the fraction of sampled actor actions outside the BC95 reference region. As can be observed, as α increases, the ratio of actions outside BC95 initially decreases, while the performance of CQL improves. This is because a moderately stronger conservative penalty effectively suppresses the Q-values of out-of-distribution actions, thereby reducing their likelihood of being selected by the policy. However, the two metrics reach their optima at different values of α: the outside-BC95 ratio reaches its minimum at $\alpha = 2$, whereas the success rate peaks at $\alpha = 10$. This discrepancy indicates that the value of α that best constrains the policy to the behavior distribution does not necessarily maximize task performance. In other words, pessimistic methods such as CQL exhibit an inherent support–value trade-off: stronger conservatism can improve distributional support, but the degree of conservatism that yields the strongest support constraint is not necessarily the one that produces the best value-based decision making.

Interestingly, when α is increased further, the outside-BC95 ratio no longer decreases monotonically, while performance begins to deteriorate. We attribute this behavior to excessive conservatism. When

the conservative penalty becomes too strong, it may suppress Q-values broadly rather than selectively penalizing unsupported actions, compressing the dynamic range of the learned value function. As a result, the critic becomes less discriminative among candidate actions, including between well-supported and poorly supported ones, which can ultimately degrade policy performance.

4 DECOUPLED POLICY EXTRACTION

Building on the notation in Sections 2.2 and 2.3, we instantiate policy extraction with a frozen BC actor and a separately learned critic. The BC actor models the dataset distribution and proposes data-supported candidate actions, while the critic is used only to rank candidates at deployment time.

Unlike previous actor-side policy improvements, where critic-derived information changes the actor parameters during training, our decoupled policy extraction paradigm applies critic information only at inference time, thereby avoiding the aforementioned OOD amplification loop problem and the support-value trade-off problem.

4.1 FROZEN BC ACTOR

We train the actor through behavior cloning. Given the offline dataset $\mathcal{D}$, the actor π_θ is optimized by maximizing the log-likelihood:

$$\mathcal{L}_{\mathrm{BC}}(\theta) = \mathbb{E}_{(s_t,g,a_t)\sim\mathcal{D}}\left[\log \pi_\theta(a_t \mid s_t, g)\right]. \tag{6}$$

For deterministic parameterizations, this reduces to supervised action regression.

After training, the actor is frozen. It receives no rewards, critic values, advantage weights, or gradients from Q_ϕ. Thus, any improvement over behavior cloning comes from critic-based selection over fixed actor samples, rather than from critic-guided updates to the actor. This also prevents critic-derived preferences from changing the proposal distribution during training, as discussed in Section 3.

4.2 CHOICE OF CRITIC

One of the advantages of decoupled policy extraction paradigm is that it substantially relaxes the requirements on critic design. In conventional offline RL, the critic is often explicitly regularized or conservatively trained to avoid assigning spuriously high values to OOD actions, since such overestimation can directly drive the actor toward unsupported regions. In our framework, however, policy improvement is performed by reranking actions proposed by a BC actor. Since the BC actor approximates the dataset distribution, the candidate set is largely restricted to data-supported regions.

As a result, the critic is no longer required to explicitly suppress arbitrary OOD actions over the entire action space, but only to provide sufficiently accurate relative value estimates among behavior-supported candidates. This substantially reduces the need for complex conservative critic designs and alleviates the support–value trade-off discussed in Section 3.2. Consequently, our framework can accommodate a broad range of critic objectives, including naive value-learning methods that do not explicitly account for OOD actions. We further show in Section 5.2 that even a vanilla Q-learning critic can achieve strong performance under our framework.

4.3 POLICY EXTRACTION VIA RERANKING

Our policy extraction is formulated as a reranking process: the BC actor proposes N data-supported action candidates, which are evaluated by the critic, and the highest-valued action is selected for execution. The candidate budget N controls the amount of inference-time search within the proposal distribution. Larger N gives the selector more candidates to compare, but can also increase exposure to ranking errors or weakly supported actions.

Therefore, we effectively replace most of the complex OOD-control machinery in coupled policy improvement methods with the tuning of a single inference-time parameter, the candidate budget N. In coupled methods, mitigating OOD overestimation requires either redesigning the method or repeatedly tuning the coefficient that balances behavioral support against value maximization. Both procedures involve costly training-time iterations. In contrast, the proposed decoupled policy extraction paradigm shifts this expensive model-level iteration to a lightweight inference-time selection problem. The candidate budget N can be adjusted without retraining either the actor or the critic, substantially reducing the cost of policy iteration.

5 EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of our proposed method and analyze the impact of its key design. The computational resources and hyper-parameter details are available in Appendix B.

5.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate six goal-conditioned continuous-control environments from OGBench (Park et al., 2025a): AntMaze-Large, HumanoidMaze-Medium, AntSoccer-Arena, Cube-Double, Scene-Play, and Puzzle-4x4. These domains span locomotion, object manipulation, contact-rich control, and state dimensions for which a single unimodal proposal can be restrictive. Each environment contributes its five fixed benchmark tasks.

Backbones. We select Q-Learning (Watkins & Dayan, 1992), IQL (Kostrikov et al., 2022), and TRL (Park et al., 2026) as representative backbones to evaluate our proposition. Q-Learning represents the most fundamental Bellman-based value learning method. IQL learns the state value function through expectile regression, enabling policy improvement without explicit maximization. TRL accelerates long-horizon value learning by exploiting value transitivity and the triangle inequality. All critic backbones are learned independently of the actor, and we use them to rerank the action candidates only. Training and evaluation details are provided in Appendix C.

5.2 MAIN RESULTS

Aggregate results are shown in Table 1. As shown in the table, using IQL as an example, the decoupled approach yields relative improvements of 38.4%, 13.9%, and 202.2% in macro-average success rate across navigation, manipulation, and logical-reasoning tasks, respectively. Taken together, these results empirically validate the effectiveness of decoupling policy improvement from actor training.

It is also worth noting that even a naive Q-learning method can serve as an effective critic for reranking. Compared with the widely adopted IQL, the decoupled paradigm based on Q-Learning achieves clear performance gains in most environments. The improvement is particularly pronounced in Puzzle, where reranking with Q-Learning achieves performance improvement of 212.8% compared with IQL. We attribute this gap to the conservative expectile value-learning of IQL. By replacing hard maximization with upper-expectile regression over in-dataset actions, IQL avoids querying potentially out-of-distribution actions but weakens the propagation of rare in-distribution high-return continuations. This effect can accumulate in long-horizon sparse-reward tasks and compress the value gap between high-quality and ordinary candidates, thereby reducing reranking accuracy. For further analysis, please refer to Section 3.

5.3 ABLATION ANALYSIS OF PERFORMANCE GAINS

To further examine the effect of actor-side policy improvement, we compare coupled and decoupled variants of FM-IQL with and without critic-based reranking on Puzzle-4x4. Table 2 reports the results across different candidate budgets N. Note that the variant *Coupled* refers to the original version of FM-IQL, which outputs only a single action at each step and does not apply reranking. Therefore, it is not applicable to settings with $N \in \{2, 4, 8, 16, 32\}$, and these entries are marked as N/A in Table 2. As shown, adding critic-based reranking to the coupled IQL policy (*Coupled + Reranking*) consistently matches or outperforms the original IQL variant across all values of N, confirming that reranking itself can provide substantial gains.

More importantly, with an appropriate candidate budget, *Decoupled + Reranking* further outperforms both the original IQL policy and its reranked counterpart, achieving the best overall performance across all methods and candidate budgets at $N = 16$. These results indicate that reranking alone remains constrained by the actor learned under the coupled training paradigm, whereas the proposed decoupled policy extraction paradigm can better exploit critic-based selection and achieve stronger overall performance.

It is also worth noting that *Coupled + Reranking* reaches its best performance at $N = 4$ and then degrades as N further increases. As discussed in Section 3.1, one potential limitation of the coupled training paradigm is that the actor distribution can be distorted by critic estimation errors, leading to an OOD amplification loop. The observed degradation at larger candidate budgets provides further empirical support for this argument.

Table 1: OGBench success rates (%; mean ± s.d. over five seeds when reported). We average the results at 800K, 900K, and 1M steps within each training seed and then report the mean and standard deviation across seeds. Candidate-budget sweeps are evaluated separately at the 1M checkpoint.

		BC	Coupled		Decoupled (Ours)		
Dataset	Task	FM-BC	FM-IQL	TRL	IQL	Transitive	Q-Learning
AntMaze-Large	task1	22 ± 10	32 ± 4	52 ± 8	59 ± 17	73 ± 7	28 ± 22
	task2	24 ± 3	29 ± 7	52 ± 12	11 ± 3	27 ± 11	23 ± 7
	task3	49 ± 10	73 ± 9	81 ± 9	81 ± 7	85 ± 8	67 ± 9
	task4	12 ± 9	10 ± 4	19 ± 2	5 ± 5	19 ± 19	23 ± 9
	task5	17 ± 3	12 ± 6	27 ± 7	15 ± 8	39 ± 11	55 ± 13
	overall	25 ± 3	31 ± 3	46 ± 3	34 ± 2	49 ± 4	41 ± 6
HumanoidMaze-Medium	task1	4 ± 3	33 ± 16	75 ± 6	81 ± 13	77 ± 14	76 ± 16
	task2	8 ± 3	55 ± 9	92 ± 4	93 ± 7	89 ± 5	88 ± 8
	task3	13 ± 6	1 ± 1	3 ± 2	19 ± 12	40 ± 11	0 ± 0
	task4	2 ± 1	0 ± 0	17 ± 8	0 ± 0	5 ± 5	0 ± 0
	task5	8 ± 3	4 ± 3	81 ± 10	57 ± 16	92 ± 8	0 ± 0
	overall	7 ± 2	19 ± 3	54 ± 3	50 ± 5	61 ± 4	33 ± 4
AntSoccer-Arena	task1	23 ± 6	88 ± 4	90 ± 3	95 ± 8	99 ± 3	95 ± 5
	task2	22 ± 6	85 ± 8	92 ± 4	99 ± 3	83 ± 5	96 ± 5
	task3	14 ± 6	90 ± 5	84 ± 3	96 ± 5	84 ± 9	83 ± 12
	task4	9 ± 2	45 ± 4	54 ± 10	81 ± 8	51 ± 21	83 ± 3
	task5	7 ± 5	83 ± 3	52 ± 3	95 ± 5	55 ± 17	89 ± 9
	overall	15 ± 3	78 ± 2	74 ± 3	93 ± 2	74 ± 4	89 ± 3
Cube-Double	task1	20 ± 7	90 ± 4	58 ± 11	96 ± 5	61 ± 11	83 ± 9
	task2	5 ± 3	90 ± 6	20 ± 11	100 ± 0	32 ± 14	77 ± 8
	task3	1 ± 2	85 ± 15	12 ± 9	100 ± 0	13 ± 4	63 ± 7
	task4	1 ± 1	16 ± 4	2 ± 2	33 ± 13	9 ± 5	12 ± 8
	task5	2 ± 2	56 ± 16	12 ± 10	65 ± 11	15 ± 11	71 ± 9
	overall	6 ± 2	67 ± 4	21 ± 8	79 ± 4	26 ± 7	61 ± 5
Scene-Play	task1	66 ± 10	100 ± 0	94 ± 3	100 ± 0	100 ± 0	85 ± 13
	task2	29 ± 17	93 ± 5	93 ± 2	100 ± 0	93 ± 6	76 ± 12
	task3	26 ± 8	99 ± 1	94 ± 5	99 ± 3	95 ± 8	92 ± 8
	task4	28 ± 4	27 ± 7	79 ± 16	37 ± 31	80 ± 9	24 ± 18
	task5	4 ± 2	1 ± 1	28 ± 10	19 ± 9	21 ± 17	13 ± 14
	overall	31 ± 6	64 ± 2	78 ± 3	71 ± 8	78 ± 4	58 ± 11
Puzzle-4x4	task1	0 ± 0	37 ± 6	44 ± 10	93 ± 6	65 ± 11	100 ± 0
	task2	0 ± 0	27 ± 9	24 ± 8	76 ± 19	43 ± 9	84 ± 7
	task3	1 ± 1	33 ± 8	39 ± 7	97 ± 3	73 ± 15	88 ± 9
	task4	0 ± 1	17 ± 2	35 ± 6	77 ± 16	45 ± 15	83 ± 14
	task5	2 ± 2	24 ± 6	33 ± 5	75 ± 11	44 ± 14	79 ± 9
	overall	1 ± 1	28 ± 4	35 ± 4	84 ± 7	54 ± 6	87 ± 6

Table 2: Overall performance of FM-IQL variants on Puzzle-4x4 across different candidate budgets N.

FM-IQL Variants	N = 1	N = 2	N = 4	N = 8	N = 16	N = 32
Coupled (Original)	32.0±3.9	N/A	N/A	N/A	N/A	N/A
Coupled + Reranking	32.0±4.7	63.6±2.7	71.6±10.6	64.9±2.3	68.9±8.3	66.2±6.7
Decoupled (Ours)	0.0±0.0	32.0±2.2	64.4±6.7	75.6±6.2	**78.2±7.7**	76.4±4.1

5.4 Further Investigations

Impact of the number of candidates N. Taking Puzzle as an example, we validate the impact of N and summarize the impact in Figure 4. As shown, performance initially improves as N increases, but declines after reaching its peak. We believe the reason is that, when N is small, increasing N allows the policy to explore the behavior-supported action region more thoroughly, thereby increasing the likelihood of identifying higher-value actions. However, as N grows further, the expected number of generated out-of-distribution actions also increases. Since such actions are often erroneously assigned high values by the Q-function, they may be preferentially selected during reranking, ultimately leading to suboptimal decisions.

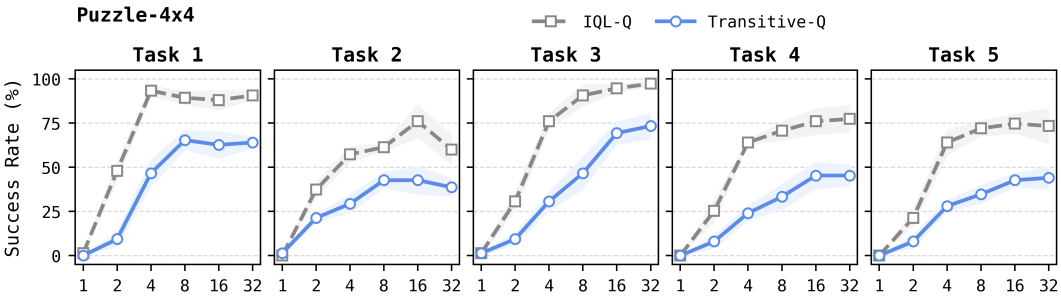


Figure 4: Impact of the number of candidates.

Inference cost. Our framework shifts critic-driven policy improvement from training-time actor optimization to inference-time candidate selection, which may introduce additional deployment cost as the candidate budget N increases. We study this trade-off on Cube-Double, a representative embodied manipulation task that requires time-sensitive online decision-making. At each decision step, all N candidates are generated in parallel through ten batched flow vector-field evaluations, followed by one batched twin-critic evaluation for reranking. Thus, increasing N widens the computation batch rather than adding serial proposer calls. As shown in Table 3, although GFLOPs increase with N, median latency remains below 0.53 ms across all configurations, while GPU memory remains nearly constant at 550–552 MiB for reranking budgets from $N = 2$ to $N = 32$.

Table 3: Inference cost of IQL-Q under different candidate budgets on Cube-Double.

N	Latency (ms)	GFLOPs	Peak mem. (MiB)
1	0.33	0.011	356
2	0.50	0.026	550
4	0.51	0.052	550
8	0.50	0.105	550
16	0.50	0.209	550
32	0.44	0.434	552

5.5 Real-world Experiments

Setup. We finetune the action head of the baseline using BC or our Decoupled Policy Extraction paradigm with pure expert demonstrations, to evaluate the effectiveness. The real-world tasks involve controlling an I7 Pro robot to perform the Pick and Install tasks. Experimental details are provided in Appendix E.

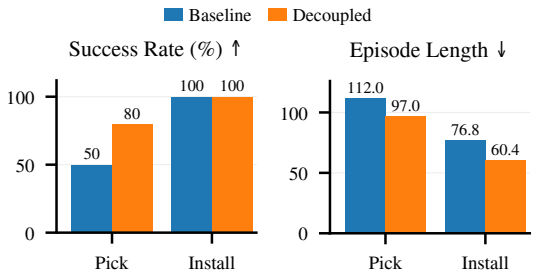


Figure 5: Real-world experimental results.

Results. The task success rates and average episode lengths are summarized in Figure 5. Following TwinRL (Xu et al., 2026), we use the average episode length, measured as the number of execution steps within an episode, to characterize task execution efficiency. A shorter episode length indicates a more direct and efficient execution. For the Pick task, despite training exclusively on expert demonstrations, the decoupled policy achieves a higher success rate while requiring 13.4% fewer inference steps on successful trials. For the Install task, although both BC and the decoupled trained baseline achieve a 100% success rate, the decoupled version reduces the number of inference steps by 21.4%. These results demonstrate that the Decoupled Policy Extraction paradigm remains effective in real-world settings trained exclusively on expert demonstrations.

6 Related Works

6.1 Actor-side policy improvement in offline RL

Offline RL learns from a fixed dataset, which mostly performs policy improvement at actor side: the method trains a parameterized actor to stay close to the behavior distribution while shifting probability toward actions assigned high value by the critic. This paradigm underlies many offline

RL methods, including behavior-constrained or regularized approaches such as BCQ, BEAR, BRAC, and TD3+BC (Fujimoto et al., 2019; Kumar et al., 2019; Wu et al., 2019; Fujimoto & Gu, 2021), conservative or implicit value-learning methods such as CQL and IQL (Kumar et al., 2020; Kostrikov et al., 2022), and actor-extraction objectives such as AWR, AWAC, CRR, and One-step RL (Peng et al., 2019; Nair et al., 2021; Wang et al., 2020; Brandfonbrener et al., 2021). Recent diffusion and flow-based offline RL methods further extend this actor-side view by injecting critic information into expressive generative policies through Q-maximization, energy guidance, distillation, or critic-gradient objectives (Wang et al., 2023; Lu et al., 2023; Park et al., 2025b; Alles et al., 2025; Li & Levine, 2026; Chen & Fleming, 2025).

6.2 Inference-time policy improvement

Some other works perform policy improvement at inference time. For instance, SfBC and EMaQ generate candidate actions from behavior models and select among them using a critic (Chen et al., 2023; Ghasemipour et al., 2021). Yet they rely on the BC actor for both proposal generation and critic learning. In contrast, our paradigm only uses the behavior model to generate candidate actions, reducing its influence on value estimation.

Generative RL methods explore another form of inference-time improvement. Diffuser guides trajectory generation using value models (Janner et al., 2022). QGF extends this idea to flow-matching policies (Zhou et al., 2026); further, GAF applies QGF-style guidance to frozen flow-matching VLA policies for embodied tasks (Yang et al., 2026). However, action generation in QGF and GAF remains coupled with the critic during inference. In contrast, our proposer generates candidates without critic intervention, and the critic is applied only for post-generation ranking.

A few prior works adopt similar decoupled structures, including V-GPS and IDQL (Nakamoto et al., 2025; Hansen-Estruch et al., 2023). Although similar, these methods focus on specific applications or algorithmic instantiations rather than systematically studying decoupling as a general policy extraction principle. We formalize this design as a decoupled policy extraction framework and analyze its mechanisms through controlled comparisons with coupled methods. Performance comparisons with similar methods can be found in Appendix F.

6.3 Diagnosing offline-RL bottlenecks

Prior work studies which component of offline RL most limits performance. Most relevantly, Park et al. (2024) identify policy extraction as a major bottleneck. We build on this perspective by opening the policy-extraction step itself: rather than treating extraction methods as monolithic rules, we explicitly control the proposer, critic, selector, and candidate budget to analyze when performance can be recovered through behavior-supported proposal and critic-based selection.

7 Conclusion and Discussion

In this work, we revisit whether learning the offline data distribution and policy improvement should be jointly carried out by the actor, and identify two fundamental limitations of coupled policy improvement offline RL methods, $i.e.$, critic-error amplification, and the support–value trade-off. To avoid these problems, we decouple policy improvement from actor training and decompose policy extraction into behavior-supported action proposal, critic evaluation, and inference-time selection. The results suggest that policy improvement is more effectively achieved by decoupling critic-based selection from actor training.

Implications for Future High-Capacity Policies. After decoupling policy improvement from actor learning, these expressive but parameter-intensive models need only be trained to model the dataset distribution through behavior cloning or supervised fine-tuning. Task-specific optimization can then be delegated to a lightweight critic that selects among actor-generated candidates at inference time, rather than repeatedly updating the actor using critic-derived objectives. This separation makes large pretrained policies substantially easier to incorporate into offline RL and allows a single proposal model to be reused across tasks by pairing it with different cheaper task-specific critics.

Limitations and Future Directions. The main limitation is proposal coverage. The critic can only select among actions generated by the behavior-cloned proposer, whose distribution is constrained by the offline dataset. Missing or extremely rare high-value behaviors therefore impose a direct performance ceiling. Increasing the candidate budget may improve coverage, but can also expose the selector to more critic ranking errors and increases inference cost.

REFERENCES

Marvin Alles, Nutan Chen, Patrick van der Smagt, and Botond Cseke. FlowQ: Energy-guided flow policies for offline reinforcement learning. *arXiv preprint arXiv:2505.14139*, 2025. URL https://arxiv.org/abs/2505.14139.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

David Brandfonbrener, William F. Whitney, Rajesh Ranganath, and Joan Bruna. Offline RL without off-policy evaluation. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/274a10ffa06e434f2a94df765cac6bf4-Abstract.html.

Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=42zs3qa2kpy.

Zhaoyang Chen and Cody Fleming. Modular diffusion policy training: Decoupling and recombining guidance and diffusion for offline RL. *arXiv preprint arXiv:2506.03154*, 2025. URL https://arxiv.org/abs/2506.03154.

Zelin Fu, Bin Tan, Changjiang Sun, Shaohui Liu, Kecheng Zheng, Yinghao Xu, Xing Zhu, Yujun Shen, and Nan Xue. Vision pretraining for dense spatial perception. *arXiv preprint arXiv:2607.05247*, 2026.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20132–20145, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/a8166da05c5a094f7dc03724b41886e5-Abstract.html.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

Seyed Kamyar Seyed Ghasemipour, Dale Schuurmans, and Shixiang Shane Gu. EMaQ: Expected-max Q-learning operator for simple yet effective offline and online RL. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3682–3691. PMLR, 2021. URL https://proceedings.mlr.press/v139/ghasemipour21a.html.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. IDQL: Implicit Q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023. URL https://arxiv.org/abs/2304.10573.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pp. 9902–9915. PMLR, 2022.

Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36: 67195–67212, 2023.

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=EblVBDNalKu.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy Q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c2073ffa77b5357a498057413bb09d3a-Abstract.html.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Qiyang Li and Sergey Levine. Q-learning with adjoint matching. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=vd4eNAdtO6.

Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22825–22855. PMLR, 2023. URL https://proceedings.mlr.press/v202/lu23d.html.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=OJiM1R3jAtZ.

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering your generalists: Improving robotic foundation models via value guidance. In *Conference on Robot Learning*, pp. 4996–5013. PMLR, 2025.

Seohong Park, Kevin Frans, Sergey Levine, and Aviral Kumar. Is value learning really the main bottleneck in offline RL? In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2509. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8ffb4e3118280a66b192b6f06e0e2596-Abstract-Conference.html.

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

Seohong Park, Qiyang Li, and Sergey Levine. Flow Q-learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48104–48127. PMLR, 2025b. URL https://proceedings.mlr.press/v267/park25f.html.

Seohong Park, Aditya Oberai, Pranav Atreya, and Sergey Levine. Transitive rl: Value learning via divide and conquer. In *International Conference on Learning Representations*, volume 2026, pp. 3217–3235, 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.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018. URL https://mitpress.mit.edu/9780262039246/reinforcement-learning/.

Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=AHvFDPi-FA.

Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S. Merel, Jost Tobias Springenberg, Scott E. Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, and Nando de Freitas. Critic regularized regression. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/588cb956d6bbe67078f29f8de420a13d-Abstract.html.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019. URL https://arxiv.org/abs/1911.11361.

Qinwen Xu, Jiaming Liu, Rui Zhou, Shaojun Shi, Nuowei Han, Zhuoyang Liu, Chenyang Gu, Shuo Gu, Yang Yue, Gao Huang, et al. Twinrl-vla: Digital twin-driven reinforcement learning for real-world robotic manipulation. *arXiv preprint arXiv:2602.09023*, 2026.

Liuhaichen Yang, Zhuang Jiang, Chenchao Sheng, and Zezhi Tang. Guided action flow: Q-guided inference for flow-matching vision-language-action policies. *arXiv preprint arXiv:2607.02092*, 2026.

Zhiyuan Zhou, Andy Peng, Charles Xu, Qiyang Li, Tobias Springenberg, Kevin Frans, and Sergey Levine. Test-time gradient guidance of flow policies in reinforcement learning. *arXiv preprint arXiv:2606.11087*, 2026.

A The Use of Large Language Models (LLMs)

We employ Large Language Models (LLMs) for grammar checking in our paper.

B Implementation Details

The optimization, architecture, goal-sampling, and evaluation settings used in our experiments are summarized below. Unless noted otherwise, each setting is shared across methods and environments.

We evaluate six standard OGBench environments with oracle goal representations. We compare a flow-matching behavior-cloning policy (FM-BC), three frozen-policy rerankers (BC+IQL-Q, BC+Q-Learning-Q, and BC+Transitive-Q), FM-IQL, and the official TRL baseline using reparameterized policy gradients.

For the standard OGBench benchmark, each training seed is evaluated at 800K, 900K, and 1M gradient updates. The frozen FM-BC and BC+Q variants use five paired training seeds: critic seed i is evaluated with the FM-BC proposal trained using seed i.

C Hyperparameter Settings

Each trainable component is optimized for 10^6 gradient steps. For each BC+Q method, we first train an FM-BC proposal policy for 10^6 steps and subsequently freeze it while training the corresponding critic for an additional 10^6 steps. In contrast, FM-IQL and TRL-RPG jointly optimize their actor and critic for 10^6 steps.

Following the notation used by OGBench and TRL, goal-sampling ratios are reported as

$$\rho = (p_{\mathrm{cur}}, p_{\mathrm{geom}}, p_{\mathrm{traj}}, p_{\mathrm{rand}}), \tag{7}$$

where p_{geom} and p_{traj} denote geometrically and uniformly sampled future trajectory goals, respectively.

Table 4: Hyperparameters shared across the standard OGBench experiments.

Hyperparameter	Value
Gradient steps per trained model	10^6
Optimizer	Adam
Learning rate	3×10^{-4}
Batch size	1024
Actor hidden dimensions	$[512, 512, 512]$
Critic/value hidden dimensions	$[512, 512, 512]$
Nonlinearity	GELU
Layer normalization	enabled for actor and critic/value networks
Critic ensemble size	2
Target-network update rate τ	0.005
Action range	$[-1, 1]$
Actor goal ratio ρ_π	$(0, 0, 1, 0)$
IQL/Q-learning goal ratio ρ_Q	$(0.2, 0.5, 0, 0.3)$
Transitive goal ratio ρ_{tr}	$(0, 1, 0, 0)$
Sparse reward for Bellman critics	0 when the sampled goal is the current state, and -1 otherwise

Table 5: Environment- and task-specific hyperparameters. The vectors contain $(N_1, \ldots, N_5)$ in the task order used by the corresponding OGBench environment. For BC+IQL-Q and BC+Transitive-Q, N is selected independently for every method and task from $\{1, 2, 4, 8, 16, 32\}$ using the five-seed mean.

Environment	γ	α_{FM}	λ_{tr}	BC+IQL-Q $N_{1:5}$	BC+Transitive-Q $N_{1:5}$
AntMaze-Large	0.99	1.0	0.0	$(16, 1, 2, 1, 1)$	$(32, 1, 2, 2, 2)$
HumanoidMaze-Med.	0.995	0.4	0.0	$(16, 8, 2, 1, 2)$	$(16, 8, 4, 4, 8)$
AntSoccer-Arena	0.99	0.4	0.5	$(8, 8, 4, 8, 8)$	$(16, 8, 8, 16, 4)$
Cube-Double	0.99	2.5	1.0	$(4, 4, 4, 16, 4)$	$(4, 4, 4, 4, 2)$
Scene-Play	0.99	2.5	1.0	$(2, 8, 4, 8, 2)$	$(8, 8, 8, 16, 16)$
Puzzle-4x4	0.99	2.5	2.0	$(4, 16, 32, 32, 16)$	$(8, 8, 32, 16, 32)$

D Representative Forms of Coupled Policy Extraction

D.1 Advantage-Weighted Regression

Advantage-weighted methods perform policy extraction by reweighting the imitation of actions contained in the offline dataset. A general objective can be written as

$$\max_\theta \ \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[w(s,a) \log \pi_\theta(a \mid s) \right], \tag{8}$$

where $w(s, a)$ is an increasing function of the estimated advantage of action a.

AWR (Peng et al., 2019) and AWAC (Nair et al., 2021) are representative examples of this formulation. In general, the weight takes the form

$$w(s,a) = f\left(A(s,a)\right), \tag{9}$$

where $A(s, a)$ is an estimated advantage and $f(\cdot)$ assigns larger weights to higher-advantage actions.

IQL (Kostrikov et al., 2022) adopts the same policy extraction principle and uses exponential advantage weighting:

$$w(s,a) = \exp\left(\beta \left(Q_\phi(s,a) - V_\psi(s)\right)\right). \tag{10}$$

In these methods, the actor is trained only on actions contained in the offline dataset, but the critic changes their relative contribution to actor optimization. Consequently, behavior fitting and policy improvement are jointly encoded into the learned policy distribution.

D.2 BC-Regularized Value Maximization

Another common form directly optimizes the actor toward actions with higher critic values while using a behavior cloning objective to constrain its deviation from the offline dataset. This class can be generally expressed as

$$\max_{\theta} \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[Q_\phi(s, \pi_\theta(s)) - \lambda \mathcal{L}_{\mathrm{BC}} \left(\pi_\theta(s), a \right) \right], \tag{11}$$

where the first term performs critic-guided policy improvement and $\mathcal{L}_{\mathrm{BC}}$ regularizes the actor toward dataset actions.

TD3+BC (Fujimoto & Gu, 2021) is a representative instance, with objective as

$$\mathcal{L}_{\mathrm{BC}} \left(\pi_\theta(s), a \right) = \| \pi_\theta(s) - a \|_2^2 . \tag{12}$$

The actor objective of TD3+BC can therefore be written as

$$\max_{\theta} \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\lambda_Q Q_\phi(s, \pi_\theta(s)) - \| \pi_\theta(s) - a \|_2^2 \right]. \tag{13}$$

Unlike advantage-weighted regression, where value information affects the relative weighting of dataset actions, BC-regularized methods allow the actor to directly optimize critic values while using behavioral regularization to control distributional deviation. Nevertheless, both forms share the same structural property: critic-guided policy improvement is incorporated into actor optimization, such that the actor simultaneously represents behavior support and value preference.

E Details of Real-World Experiments

Baseline descriptions. For the real-world evaluation, we construct our policy backbone by combining lingbot-vision-vit-large (Fu et al., 2026) as the vision encoder with the action-expert architecture of GR00T N1.7 (Bjorck et al., 2025). The same action expert is used for both BC and our decoupled method: BC directly executes actions generated by the expert, while our method treats the frozen expert as a proposal policy and uses a separately learned critic to rerank multiple candidate actions at inference time.

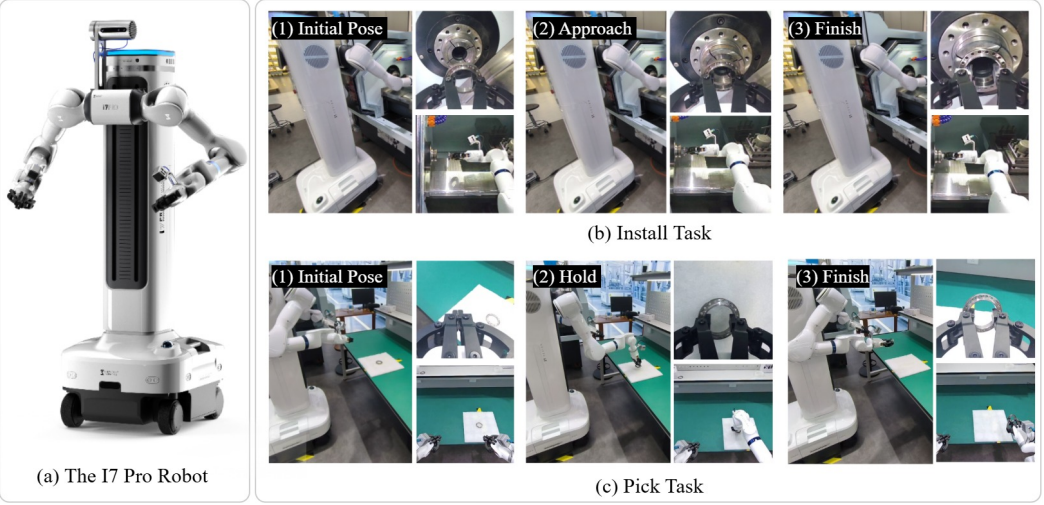
Task descriptions. We conduct the real-world evaluation on the I7 Pro, a mobile manipulation platform equipped with dual robotic arms and onboard visual sensing, as shown in Figure 6. We consider two manipulation tasks in a CNC workspace: *Pick* and *Install*. In the Pick task, the robot approaches the target workpiece, holds it, and retrieves it from the designated position. In the Install task, the robot starts with the workpiece already grasped, approaches the target installation position, and performs precise alignment and insertion. For each task and method, we conduct 20 evaluation trials to report the success rate and the average episode length. Compared with Pick, Install requires substantially finer manipulation, as the robot must precisely align the workpiece with the target before completing the insertion. Representative execution stages and corresponding close-up views are shown in Figure 6.

F Performance Comparison with similar methods

Some methods adopt related ideas, to varying degrees, to achieve inference-time policy improvement, as discussed in Section 6. We further compare our decoupled policy extraction framework with these related methods, with the results summarized in Table 6.

G Actor-Drift Diagnostic: Implementation Details

This appendix provides the complete actor objectives and reproducibility details for the actor-drift diagnostic in Figure 2. All actions shown in the diagnostic are generated directly by the corresponding diffusion actor, without candidate reranking or Q-dependent inference.



(a) The I7 Pro Robot

(b) Install Task

(c) Pick Task

Figure 6: Real-world experimental platform and manipulation tasks. For each task, three representative execution stages are shown. Each stage is illustrated with a global view of the robot and its workspace, along with close-up views that provide finer details of the manipulation process and the wrist-camera observation.

Methods. For EDP-BC, EDP-CRR, and EDP-TD3, we use a common Efficient Diffusion Policy (EDP) (Kang et al., 2023) backbone, trained with behavior cloning alone, behavior cloning augmented with CRR-style advantage-weighted regression, and behavior cloning augmented with TD3-style Q-value maximization, respectively.

Actor objectives. All variants use the same diffusion actor and behavior-cloning loss $\mathcal{L}_{\text{diff}}$. Their objectives are

$$\mathcal{L}_{\text{actor}} = \mathcal{L}_{\text{diff}} + \lambda \mathcal{L}_{\text{guide}}, \tag{14}$$

where

$$\mathcal{L}_{\text{guide}}^{\text{CRR}} = -\mathbb{E}[\text{sg}(w(A_{\text{CRR}})) \log \mathcal{N}(a; \hat{a}_\theta, I)], \tag{15}$$

$$\mathcal{L}_{\text{guide}}^{\text{TD3}} = -\mathbb{E}\left[\frac{\alpha_{\text{TD3}} Q_j(s, \hat{a}_\theta, g)}{\text{sg}(\mathbb{E}_{\mathcal{B}}\,|Q_j(s, \hat{a}_\theta, g)|)}\right]. \tag{16}$$

Here, $\hat{a}_\theta$ is the reconstructed clean action, $j \sim \text{Uniform}\{1, 2\}$, and sg denotes stop-gradient. For CRR,

$$w(A) = \min\{w_{\max}, \exp(A/\beta)\}, \tag{17}$$

where A_{CRR} compares the dataset-action value against a policy-sampled value baseline. EDP-BC sets $\mathcal{L}_{\text{guide}} = 0$.

PCA and BC95 construction. A separate BC reference cloud is generated from an independent set of 1000 initial noise vectors, which is reused across checkpoints. At each checkpoint, the first 500 projected BC actions form the KDE-fit split and the remaining 500 form the conformal-calibration split. The KDE-fit actions from all four checkpoints are pooled to fit one shared PCA basis.

For each checkpoint, we fit a two-dimensional Gaussian KDE with Scott bandwidth and full covariance. The nonconformity score is the negative log KDE density. With 500 calibration actions, the BC95 threshold is the 476-th smallest calibration score, $\lceil (500 + 1) \times 0.95 \rceil = 476$. An action is classified as outside BC95 when its split-conformal p-value is at most 0.05. This classification measures coverage relative to the same-checkpoint BC distribution in the shared PCA plane and is not a formal environment-level OOD test.

Visualization. All panels use the same PCA basis, but their axis limits are adjusted independently to contain the complete action cloud and BC95 contour. Points inside and outside BC95 use identical

Table 6: OGBench success rates (%; mean ± s.d. over five seeds when reported). We average the results at 800K, 900K, and 1M steps within each training seed and then report the mean and standard deviation across seeds.

Dataset	Task	Related Approaches			Our Decoupled Paradigm		
		IDQL	V-GPS	FlowQ	IQL	Transitive	Q-Learning
AntMaze-Large	task1	44 ± 10	11 ± 12	3 ± 6	59 ± 17	73 ± 7	28 ± 22
	task2	0 ± 0	1 ± 3	7 ± 8	11 ± 3	27 ± 11	23 ± 7
	task3	68 ± 12	68 ± 6	5 ± 9	81 ± 7	85 ± 8	67 ± 9
	task4	0 ± 0	11 ± 9	0 ± 0	5 ± 5	19 ± 19	23 ± 9
	task5	0 ± 0	15 ± 9	0 ± 0	15 ± 8	39 ± 11	55 ± 13
	overall	22 ± 4	21 ± 4	3 ± 4	34 ± 2	49 ± 4	41 ± 6
HumanoidMaze-Medium	task1	68 ± 14	16 ± 10	7 ± 8	81 ± 13	77 ± 14	76 ± 16
	task2	84 ± 14	25 ± 12	17 ± 13	93 ± 7	89 ± 5	88 ± 8
	task3	0 ± 0	24 ± 8	16 ± 9	19 ± 12	40 ± 11	0 ± 0
	task4	0 ± 0	1 ± 3	11 ± 9	0 ± 0	5 ± 5	0 ± 0
	task5	3 ± 4	21 ± 13	21 ± 9	57 ± 16	92 ± 8	0 ± 0
	overall	31 ± 5	18 ± 3	14 ± 3	50 ± 5	61 ± 4	33 ± 4
AntSoccer-Arena	task1	95 ± 6	17 ± 4	17 ± 6	95 ± 8	99 ± 3	95 ± 5
	task2	96 ± 4	15 ± 11	5 ± 7	99 ± 3	83 ± 5	96 ± 5
	task3	92 ± 7	25 ± 10	5 ± 6	96 ± 5	84 ± 9	83 ± 12
	task4	80 ± 15	8 ± 6	5 ± 9	81 ± 8	51 ± 21	83 ± 3
	task5	83 ± 10	12 ± 7	3 ± 4	95 ± 5	55 ± 17	89 ± 9
	overall	89 ± 3	15 ± 3	7 ± 2	93 ± 2	74 ± 4	89 ± 3
Cube-Double	task1	59 ± 20	63 ± 15	59 ± 20	96 ± 5	61 ± 11	83 ± 9
	task2	59 ± 20	0 ± 0	9 ± 9	100 ± 0	32 ± 14	77 ± 8
	task3	48 ± 19	0 ± 0	0 ± 0	100 ± 0	13 ± 4	63 ± 7
	task4	8 ± 6	0 ± 0	0 ± 0	33 ± 13	9 ± 5	12 ± 8
	task5	21 ± 17	1 ± 3	3 ± 4	65 ± 11	15 ± 11	71 ± 9
	overall	39 ± 12	13 ± 3	14 ± 5	79 ± 4	26 ± 7	61 ± 5
Scene-Play	task1	100 ± 0	88 ± 11	91 ± 4	100 ± 0	100 ± 0	85 ± 13
	task2	96 ± 4	35 ± 14	35 ± 14	100 ± 0	93 ± 6	76 ± 12
	task3	89 ± 8	4 ± 6	28 ± 15	99 ± 3	95 ± 8	92 ± 8
	task4	21 ± 32	0 ± 0	8 ± 9	37 ± 31	80 ± 9	24 ± 18
	task5	9 ± 10	0 ± 0	1 ± 3	19 ± 9	21 ± 17	13 ± 14
	overall	63 ± 8	25 ± 5	33 ± 2	71 ± 8	78 ± 4	58 ± 11
Puzzle-4x4	task1	96 ± 6	75 ± 14	0 ± 0	93 ± 6	65 ± 11	100 ± 0
	task2	56 ± 16	52 ± 21	0 ± 0	76 ± 19	43 ± 9	84 ± 7
	task3	85 ± 13	61 ± 14	1 ± 3	97 ± 3	73 ± 15	88 ± 9
	task4	75 ± 17	51 ± 6	0 ± 0	77 ± 16	45 ± 15	83 ± 14
	task5	81 ± 9	55 ± 14	0 ± 0	75 ± 11	44 ± 14	79 ± 9
	overall	79 ± 8	59 ± 9	0 ± 1	84 ± 7	54 ± 6	87 ± 6

size, opacity, color mapping, and draw order. The dashed BC95 contour is rendered above the action samples.

CRR actions are colored by the diagnostic advantage

$$A_{\text{diag}}(s, a, g) = Q_{\min}(s, a, g) - \frac{1}{K} \sum_{k=1}^{K} Q_{\min}(s, \tilde{a}_k, g), \tag{18}$$

where $\tilde{a}_k$ are sampled from the current CRR policy. One robust color range is shared across all CRR checkpoints. TD3 actions are colored by the within-checkpoint percentile rank of $Q_{\min} = \min(Q_1, Q_2)$; consequently, TD3 colors indicate within-panel rankings and should not be compared as absolute Q-values across checkpoints.

Table 7: Hyperparameters for the diffusion-policy actor-drift diagnostic on AntSoccer-Arena.

Hyperparameter	Value
Shared training settings	
Environment	antsoccer-arena-navigate-oraclerep-v0
Gradient steps	10^6 per method
Optimizer	AdamW
Learning rate	3×10^{-4}
Batch size	1024
Diffusion actor	3×512 MLP, Mish activation, LayerNorm enabled
Actor-goal probabilities	$(p_{\mathrm{cur}}, p_{\mathrm{traj}}, p_{\mathrm{rand}}) = (0, 1, 0)$, uniform trajectory-goal sampling
Diffusion timesteps / schedule	100 / linear
Time-embedding dimension	16
Inference sampler	deterministic DDIM, 15 steps, $\eta = 0$
Diffusion-loss coefficient	$c_{\mathrm{diff}} = 1.0$
Action range	$[-1, 1]$
Coupled-critic settings: CRR and TD3 only	
Critic	double Q, 3×512 MLP
Discount factor	$\gamma = 0.99$
Target-network update	$\tau = 0.005$, every 5 updates after update 1000
Critic-goal probabilities	$(p_{\mathrm{cur}}, p_{\mathrm{traj}}, p_{\mathrm{rand}}) = (0.2, 0.5, 0.3)$, uniform trajectory-goal sampling
Sparse critic reward	0 for the current-state goal and -1 otherwise
Method-specific settings	
EDP-BC	no critic or guide loss
EDP-CRR	$\beta = 15$, $w_{\max} = 20$, $K = 10$, baseline $\sigma = 0.5$, $c_{\mathrm{guide}} = 0.3$, critic LayerNorm enabled
EDP-TD3	$\alpha_{\mathrm{TD3}} = 1.5$, $c_{\mathrm{guide}} = 1.0$, critic LayerNorm disabled

H CQL Conservatism Diagnostic

This appendix describes the goal-conditioned CQL diagnostic reported in Figure 3. Experiments are conducted on `cube-double-play-oraclerep-v0` using a tanh-Gaussian actor and a double critic. The critic augments the standard Bellman objective with an importance-corrected CQL log-sum-exp penalty weighted by α_{CQL}. The penalty uses 10 samples from the uniform action distribution, the current policy, and the policy at the next observation. The SAC actor objective is held fixed across α_{CQL}, so conservatism affects the policy only through the learned critic.

BC95 construction. For each repeat, we uniformly sample 32 valid dataset trajectories without replacement. Each anchor consists of the midpoint nonterminal state of one trajectory and its terminal oracle goal. We generate 1000 actions from the evaluated CQL actor and an independent 1000-action reference cloud from a behavior-cloning actor trained for 10^6 updates.

For every anchor, the first 500 BC actions fit a two-dimensional PCA projection and a Gaussian KDE with Scott bandwidth. The remaining 500 actions form the split-conformal calibration set. Using negative log KDE density as the nonconformity score R. An action is counted as outside BC95 when $p(a) \leq 0.05$. This procedure is repeated five times using matched anchor and sampling seeds. Outside BC95 measures deviation from a same-state BC reference in the two-dimensional PCA plane; it is not a formal environment-level OOD test.

Training and aggregation. All policies are trained for 10^6 gradient updates. Success is evaluated deterministically with evaluation seed 0. When available, the 800K, 900K, and 1M checkpoints are first averaged within each model. Outside BC95 is averaged over anchors, the five conformal repeats, selected checkpoints, and finally the available model slots.

Table 8: Shared settings for the CQL conservatism diagnostic.

Setting	Value
Environment	`cube-double-play-oraclerep-v0`
Gradient updates	10^6
Optimizer / learning rate	Adam / 3×10^{-4}
Batch size	1024
Actor	3×512 MLP, LayerNorm, tanh output
Critic	Double Q, 3×512 MLP, LayerNorm
Discount / target update	$\gamma = 0.99$, $\tau = 0.005$
CQL temperature	1.0
Proposal samples	10 per proposal source
Actor goal probabilities	$(0, 1, 0)$
Critic goal probabilities	$(0.2, 0.5, 0.3)$