ENSEMBLE VARIANCE SUPPRESSION CRITIC FOR OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline reinforcement learning (RL) enables learning policies from fixed datasets without costly or unsafe environment interactions. To mitigate value overestimation in this setting, ensemble critics are widely used and have demonstrated remarkable effectiveness. However, we reveal a critical flaw in existing methods: the ensemble variance of Q-values often exceeds the action-value variance conditioned on the same state. This over-dispersion obscures true action advantages and degrades the identification capability for out-of-distribution (OOD) actions, causing policies to chase ensemble noise and resulting in sub-optimal policy optimization. To address this, we propose an In-Distribution Variance Suppression Critic. Specifically, we introduce a variance suppression term that explicitly penalizes ensemble disagreement on in-distribution dataset points, thereby aligning the ensemble uncertainty with the intrinsic action-value variation. Combined with EDAC, an ensemble-based offline RL baseline, our method consistently achieves competitive or stronger results among model-free offline RL algorithms across multiple diverse benchmarks.

1 INTRODUCTION

Offline reinforcement learning (RL) aims to learn a policy from a fixed, previously collected dataset without requiring any further interaction with the environment. This paradigm is appealing for real-world applications where online interaction is expensive, slow, or unsafe, such as robotics, healthcare, and autonomous driving (Levine et al., 2020; Prudencio et al., 2023; Lange et al., 2012).

The central challenge in offline RL is the distribution shift between the behavior policy that collected the dataset and the policy being learned. When the learned policy queries actions that lie outside the support of the dataset (out-of-distribution, OOD), the Q-function must extrapolate, and value estimates on these OOD actions are systematically overestimated. Maximizing such an overestimated Q-function then yields a poor policy. To mitigate this, prior work constrains the learned policy to remain close to the dataset (Fujimoto et al., 2019; Kumar et al., 2019; Wu et al., 2019; Fujimoto & Gu, 2021), learns conservative value functions (Kumar et al.; Kostrikov et al., 2022), or relies on conservative model-based rollouts (Yu et al., 2020; Kidambi et al., 2020; Yu et al., 2021; Rigter et al., 2022).

Among these, ensemble-based critics form a prominent and empirically strong family. The key idea is to use the disagreement (variance) among an ensemble of Q-functions as a proxy for epistemic uncertainty, and then penalize actions on which the ensemble members disagree, thereby discouraging OOD actions. EDAC (An et al., 2021) achieves state-of-the-art performance by combining a large diversified Q-ensemble with a gradient-diversity regularization term, and SAC-N-BC (Beeson & Montana, 2024) and related works (Wang et al., 2022) follow the same principle. All of these methods rest on an implicit assumption: ensemble disagreement is a faithful signal of OOD-ness, and larger disagreement is unambiguously better.

In this paper, we reveal that this assumption does not hold. We observe that the ensemble variance of Q-values systematically exceeds the intrinsic action-value variance conditioned on the same state—even on in-distribution dataset points. This over-dispersion has two harmful consequences. First, the policy gradient chases ensemble noise rather than the true action advantage, leading to sub-optimal policy optimization. Second, the inflated disagreement blurs the boundary between in-distribution

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.

and OOD actions, degrading the ability of the ensemble to identify OOD actions in the first place (Figure 1).

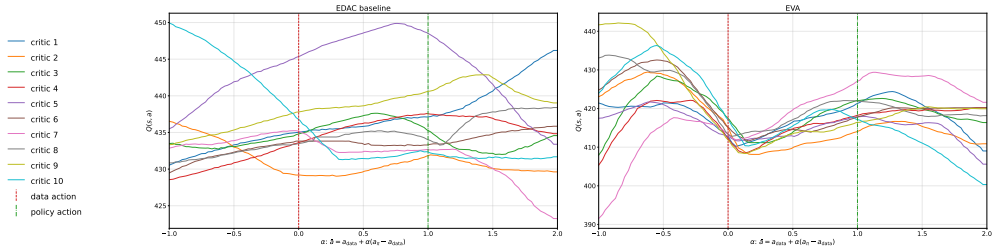


Figure 1: Motivating evidence of ensemble over-dispersion (halfcheetah-medium-expert-v2, one state s). Left: colour legend for the ten ensemble critics. Middle: $Q(s, \bar{a})$ as a function of the interpolation coefficient α, where $\bar{a} = a_{\text{data}} + \alpha(a_\pi - a_{\text{data}})$, for the ten critics of the EDAC baseline (one curve per critic); the red dashed line marks the dataset action ($\alpha = 0$) and the green dash-dotted line marks the policy action ($\alpha = 1$). Right: the same for our variance-suppressed critic (EVA).

As shown in the middle panel, the EDAC ensemble members disagree widely, and their spread on in-distribution dataset points is as large as it is away from them; disagreement therefore does not separate OOD actions from dataset ones, and the policy is left chasing ensemble noise. Suppressing the in-distribution ensemble variance (right panel) tightens the spread while preserving the increase towards the policy action, making the true action advantage—and with it the OOD boundary—visible.

To address this, we propose EVA (Ensemble VAriance suppression), an in-distribution variance suppression critic. EVA introduces a simple variance-suppression term that explicitly penalizes ensemble disagreement on in-distribution dataset points, thereby re-aligning the ensemble uncertainty with the intrinsic action-value variation. The method is simple, model-free, and can be readily combined with existing ensemble-based offline RL algorithms. We instantiate EVA on top of EDAC for the standard D4RL benchmarks, and on top of Flow Q-Learning (FQL) for the scalable offline RL benchmark OGBench.

Experiments across diverse benchmarks demonstrate consistent improvements. On D4RL MuJoCo locomotion and AntMaze tasks (Fu et al., 2020), EVA combined with EDAC achieves competitive or stronger results than recent state-of-the-art model-free methods, including EPIC (Zhang et al., 2026), FANS (Wang et al., 2026), and SAC-N-BC (Beeson & Montana, 2024). On the challenging OGBench tasks (Park et al., 2025a), adding an ensemble critic equipped with EVA to FQL (Park et al., 2025b) yields significant gains over FQL, Q-Flow (Doo et al., 2026), and FAC (Chae et al., 2026). Notably, we are the first to experimentally verify that a simple ensemble-based critic can bring large gains to scalable offline RL, providing a strong and easily extendable foundation for future research.

Our contributions are summarized as follows:

- We reveal a critical flaw of existing ensemble-based offline RL methods: the ensemble variance is over-dispersed and exceeds the intrinsic action-value variance on in-distribution data, which degrades both policy optimization and OOD identification.
- We propose EVA, a simple and effective in-distribution variance-suppression term that re-aligns ensemble uncertainty with intrinsic action-value variation, and can be combined with existing ensemble critics.
- We show that EVA consistently improves EDAC on D4RL and FQL on OGBench, achieving competitive or stronger results against state-of-the-art model-free and scalable offline RL methods.
- As a side contribution, we provide the first experimental evidence that a simple ensemble critic substantially boosts scalable offline RL.

2 RELATED WORK

2.1 OFFLINE REINFORCEMENT LEARNING

Offline RL learns a policy from a static dataset without environment interaction. The main difficulty is value overestimation on OOD actions induced by distribution shift. Existing remedies fall into several categories. Policy-constraint methods restrict the learned policy to remain close to the behavior policy, for instance via behavior cloning (TD3+BC (Fujimoto & Gu, 2021)), support matching (BCQ (Fujimoto et al., 2019)), or divergence constraints (BEAR (Kumar et al., 2019), BRAC (Wu et al., 2019)). Conservative value methods directly penalize overestimated values (CQL (Kumar et al.)) or avoid querying OOD actions through expectile regression (IQL (Kostrikov et al., 2022)). Model-based methods learn a dynamics model and impose conservatism on synthetic rollouts (MOPO (Yu et al., 2020), MOReL (Kidambi et al., 2020), COMBO (Yu et al., 2021), RAMBO (Rigter et al., 2022)). Sequence modeling casts offline RL as conditional generation (Decision Transformer (Janner et al., 2021)). Our work is orthogonal to these directions: it operates directly on the ensemble critic and does not modify the policy constraint, the value penalty, or the dynamics model.

2.2 UNCERTAINTY-BASED ENSEMBLE METHODS

Ensembles have long been used to quantify uncertainty in deep RL; in offline RL, this idea is adopted to detect and penalize OOD actions. SAC-N uses a large ensemble and takes the minimum over members as a pessimistic estimate. EDAC (An et al., 2021) further regularizes the ensemble with a gradient-diversity term so that members disagree more on OOD actions. SAC-N-BC (Beeson & Montana, 2024) studies the trade-off between policy constraint and ensemble size, and Bootstrapped Transformer (Wang et al., 2022) extends uncertainty-based offline RL to transformers. A common thread of these methods is to amplify or exploit ensemble diversity as a signal of OOD-ness. In contrast, we show that ensemble diversity is already over-dispersed on in-distribution data, and that explicitly suppressing in-distribution ensemble variance—rather than amplifying it—improves performance. Our variance-suppression term is thus complementary to existing diversity-based regularizers and can be combined with them, as we do with EDAC.

Ensemble critics descend from double Q-learning (Mnih et al., 2013), whose twin-critic minimum target underlies TD3 (Fujimoto et al., 2018) and SAC (Haarnoja et al., 2018). Rather than enlarging or diversifying the ensemble, we suppress the in-distribution variance that is already present, sharpening the OOD signal these methods rely on.

2.3 SCALABLE OFFLINE RL

Recent work has shifted attention toward scalable offline RL, which targets large, diverse, goal-conditioned datasets, with OGBench (Park et al., 2025a) as a standard benchmark. Flow-policy methods—FQL (Park et al., 2025b), Q-Flow (Doo et al., 2026), FAC (Chae et al., 2026), QIPO-OT (Zhang et al., 2025), IFQL (Park et al., 2025b), and FAWAC (Park et al., 2025b)—and diffusion-policy methods—IDQL (Hansen-Estruch et al., 2023), SRPO (Chen et al., 2024), and CAC (Ding & Jin, 2024)—largely rely on a single or a pair of critics rather than large ensembles. We show that augmenting FQL with a variance-suppressed ensemble critic yields consistent gains on OGBench, suggesting that ensemble-based uncertainty estimation remains underexplored in this scalable, expressive-policy regime.

3 METHOD

Current ensemble-based methods have large ensemble variance, which not only makes the policy gradient tend to chase noise instead of the real action advantage but also makes it hard for the policy to tell the difference between in-distribution actions and out-of-distribution actions. We suppress the variance of Q values within the dataset by a simple and effective EVA loss. The EVA loss is as follow:

$$\mathcal{L}_{\text{EVA}}(Q_\Theta) := \mathbb{E}_{(s,a)\sim\mathcal{D},1\le j\neq i\le N}[(Q(\theta_i,s,a)-Q(\theta_j,s,a))^2]. \tag{1}$$

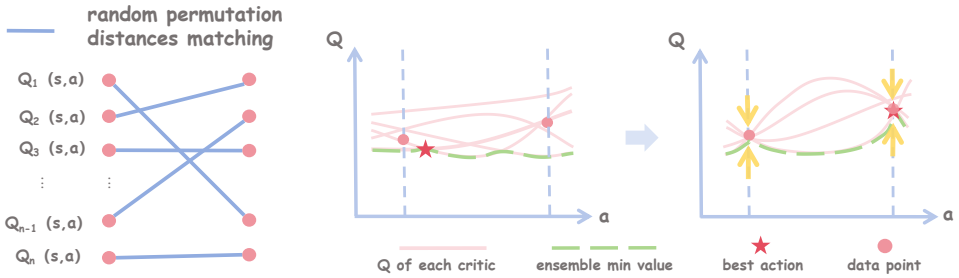


Figure 2: Overview of the proposed method: the ensemble critic is trained with the EVA loss that pulls the Q-values of different ensemble members together on in-distribution data, suppressing their variance.

In the equation, n represents the size of the critic ensemble. As illustrated in Figure 2. In practice, we calculate this term by closing the distance between all the ensemble members and a random permutation of themselves, as illustrated in Figure 2. For details, see Equation 5

Following the EDAC paper (An et al., 2021), the EDAC loss would be:

$$\mathcal{L}_{edac}(Q_\Theta) := \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}\left[\frac{1}{N-1}\sum_{1\le i\ne j\le N}\langle\nabla_{\mathbf{a}}Q(\theta_i,\mathbf{s},\mathbf{a}),\,\nabla_{\mathbf{a}}Q(\theta_j,\mathbf{s},\mathbf{a})\rangle\right], \tag{2}$$

where $\langle a, b\rangle$ represents cosine similarity.

The critic loss is:

$$\mathcal{L}_{critic}(Q_\Theta) := \mathcal{L}_{td}(Q_\Theta) + \eta\mathcal{L}_{edac}(Q_\Theta) + \beta\mathcal{L}_{\text{EVA}}(Q_\Theta), \tag{3}$$

where η and β is hyper parameters controlling the weight of each loss part.

4 Practical Implementation

Note that the sum over all pairs in equation 1 is equal to the sample variance of the ensemble in theory:

$$\frac{1}{\binom{N}{2}}\sum_{1\le i<j\le N}\left(Q(\theta_i,s,a)-Q(\theta_j,s,a)\right)^2 = 2\cdot\frac{1}{N-1}\sum_{i=1}^{N}\left(Q(\theta_i,s,a)-\bar{Q}(s,a)\right)^2 = 2s^2, \tag{4}$$

where $\bar{Q}(s,a) = \frac{1}{N}\sum_{i=1}^{N}Q(\theta_i,s,a)$ and s^2 is the sample variance. In practice, however, we do not directly minimize the full pairwise variance; instead we decompose it into a random-permutation form like equation 1. Within each batch, every ensemble member's Q value is pulled towards the Q-value of another randomly member's Q value,

$$\mathcal{L}_{\text{EVA}}(Q_\Theta) := \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\frac{1}{N}\sum_{i=1}^{N}\left(Q(\theta_i,s,a)-Q(\theta_{\sigma(i)},s,a)\right)^2\right], \tag{5}$$

where σ is a random permutation of $\{1,\ldots,N\}$ with $\sigma(i)\ne i$, resampled every batch. This stochastic pairing injects randomness that prevents the ensemble members from collapsing into a homogeneous set and getting stuck in the same local optimum. Figure 2 intuitively summarizes the practical approach.

The critic ensemble can compute the TD target in two ways. EDAC (An et al., 2021) aggregates by taking the minimum over the ensemble, so that all critics share a single target,

$$y_i = r + \gamma\min_{1\le i\le N}Q(\theta_i^-,s',a'), \tag{6}$$

whereas MSG (Ghasemipour et al., 2022) performs no aggregation, letting each critic compute its own TD error independently,

$$y_i = r + \gamma Q(\theta_i^-, s', a'), \quad i = 1, \ldots, N. \tag{7}$$

The temporal difference error is then computed as the mean squared error between each critic's prediction and its target,

$$\mathcal{L}_{td}(Q_\Theta) := \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}} \left[\frac{1}{N} \sum_{i=1}^{N} \left(Q(\theta_i, s, a) - y_i \right)^2 \right], \tag{8}$$

where y_i is defined by either equation 6 or equation 7.

The actor is optimized by minimizing the aggregated Q-value together with a behavior-cloning regularization, following SAC-N-BC (Beeson & Montana, 2024):

$$\mathcal{L}_\pi := -\mathbb{E}_{s\sim\mathcal{D}} \left[\min_{1\le i\le N} Q(\theta_i, s, \pi(s)) \right] + \alpha\, \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\|a - \pi(s)\|_2^2 \right], \tag{9}$$

where α controls the strength of the behavior-cloning term.

We use the minimum aggregation on D4RL MuJoCo (following EDAC) and the MSG aggregation on D4RL AntMaze and the OGBench AntSoccer and cube-double tasks (Ghasemipour et al., 2022). For the behavior-cloning strength, $\alpha = 0$ on D4RL MuJoCo (i.e., pure EDAC), while on D4RL AntMaze we follow the SAC-N-BC hyperparameters and on OGBench we follow the FQL (Park et al., 2025b) hyperparameters.

5 EXPERIMENTS

We organize the experiments around four questions. First, does EVA consistently improve strong ensemble-based baselines across diverse offline RL benchmarks, from the standard D4RL tasks to the scalable OGBench tasks? We answer this in Section 5.1. Second, what is the additional computational cost of the variance-suppression term, and how does it scale with the ensemble size? We study this in Section 5.2. Third, how sensitive is the method to its key design choices, namely the suppression strength β, the ensemble size n, and the EDAC subset size k? We address this in Section 5.3. Fourth, why does EVA improve performance in the first place? We analyze the underlying mechanism in Section 5.3.

5.1 MAIN RESULTS

We tested the effectiveness of our method in the Mujoco locomotion (Todorov et al., 2012) and Antmaze (Fu et al., 2020) tasks of D4RL (Fu et al., 2020), as well as the AntSoccer, cube-double, and humanoidmaze-large environments of OGBench (Park et al., 2025a), which are among the most challenging and algorithm-discriminating tasks in that benchmark.

5.1.1 MUJOCO EXPERIMENTS

We implement our method based on the EDAC (An et al., 2021) code of CORL (Tarasov et al., 2022) codebase. We compare our method to the previous state-of-the-art model-free offline RL algorithms EPIC (Zhang et al., 2026), FANS (Wang et al., 2026), and SAC-N-BC (Beeson & Montana, 2024) on the D4RL tasks, together with the additional AntMaze baselines MCQ (Lyu et al., 2022), ReBRAC (Tarasov et al., 2023), EPQ (Yeom et al., 2024), and SAC-RND (Nikulin et al., 2023). Following the standard evaluation protocol of prior work (Fu et al., 2020; Kumar et al.; Kostrikov et al., 2022; An et al., 2021; Fujimoto & Gu, 2021), the reported score of each run is the evaluation score at the final training step. The results are presented in table 1 and 2. The default and task-specific hyperparameters are reported in Appendix A.1.

On D4RL MuJoCo, EDAC+EVA attains a top-two score on five of the nine tasks and the highest average (96.2), surpassing EDAC (92.9) as well as the strong uncertainty-based baselines Epic (95.7) and FANS (96.0). The gains are most pronounced on the medium-replay tasks, where the dataset is narrow and over-dispersion is hardest to anchor: for example, on halfcheetah-medium-replay our

Table 1: Normalized average score of EDAC+EVA (ours) and baselines on D4RL MuJoCo tasks. Each score is the mean over 4 random seeds. The best two results per row are bolded.

	Diffusion Policies			Flow Policies		Gaussian Policies						
	Normal (ensemble_size ≤ 2)			Normal (ensemble_size ≤ 2)		Normal (ensemble_size ≤ 2)			Uncertainty-based (ensemble_size ≥ 2)			
Dataset	**IDQL**	**SRPO**	**CAC**	**FQL**	**QIPO-OT**	**IQL**	**CQL**	**TD3+BC**	**EDAC**	**Epic**	**FANS**	**EDAC+EVA(ours)**
halfcheetah-m	51.0	60.4	**69.1**	60.3	54.2	47.4	44.0	48.3	67.7	**68.9**	66.6	68.2±1.9
hopper-m	65.4	95.5	80.7	68.1	94.1	66.3	58.5	59.3	101.7	**102.2**	**104.6**	102.0±0.4
walker2d-m	82.5	84.4	83.1	77.2	87.6	78.3	72.5	83.7	93.4	95.9	**101.0**	**99.1±2.6**
halfcheetah-m-r	45.9	51.4	58.7	49.3	48.0	44.2	45.5	44.6	62.1	**64.8**	55.9	**66.4±1.1**
hopper-m-r	92.1	101.2	99.7	49.8	101.3	94.7	95.0	60.9	99.7	102.0	**103.2**	**104.1±0.7**
walker2d-m-r	85.1	84.6	79.5	53.1	78.6	73.9	77.2	81.8	87.1	95.2	**98.3**	**95.9±1.8**
halfcheetah-m-e	95.9	92.2	84.3	99.6	94.5	86.7	91.6	90.7	**104.8**	103.8	102.8	**106.1±1.8**
hopper-m-e	108.6	100.1	100.4	83.1	108.0	91.5	105.4	98.0	105.2	**112.3**	**113.3**	111.1±0.0
walker2d-m-e	112.7	114.0	110.4	106.1	110.9	109.6	108.8	110.1	114.8	**116.4**	**118.1**	113.1±0.7
Avg	82.1	87.1	85.1	71.8	86.4	77.0	77.6	75.3	92.9	95.7	**96.0**	**96.2**

Table 2: Average success rate (%) of EDAC+EVA (ours) and baselines on D4RL AntMaze tasks. Each score is the mean over 4 random seeds. The best result per row is bolded.

	Diffusion Policies			Flow Policies		Gaussian Policies							
	Normal (ensemble_size ≤ 2)			Normal (ensemble_size ≤ 2)		Normal (ensemble_size ≤ 2)					Uncertainty-based (ensemble_size ≥ 2)		
Dataset	**IDQL**	**SRPO**	**CAC**	**QIPO-OT**	**FQL**	**TD3+BC**	**IQL**	**CQL**	**MCQ**	**ReBRAC**	**EPQ**	**SAC-RND**	**EDAC+EVA(ours)**
am-m-pl-v2	84.5	80.7	56.8	80.0	78.0	10.6	71.2	61.2	52.5	84.0	85.0	38.5	**90±1**
am-m-dv-v2	84.8	75.0	0.0	86.4	71.0	3.0	70.0	53.7	37.5	76.3	86.7	74.7	**89±2**
am-l-pl-v2	63.5	53.6	0.0	55.5	84.0	0.2	39.6	15.8	2.5	60.4	40.0	43.9	**85±2**
am-l-dv-v2	67.9	53.6	0.0	32.1	**83.0**	0.0	47.5	14.9	7.5	54.4	36.7	45.7	77±3
Avg	75.2	65.7	14.2	63.5	79.0	3.4	57.1	36.4	25.0	68.8	62.1	50.7	**85**

method reaches 66.4 versus 62.1 for EDAC, and on hopper-medium-replay it reaches 104.1 versus 99.7. This is consistent with our motivation, since suppressing in-distribution ensemble variance helps most precisely when there is little data to anchor the ensemble's uncertainty. On the sparse-reward, long-horizon AntMaze tasks, EDAC+EVA achieves the best score on three of the four tasks and the highest average (85), clearly above the next-best baselines (FQL 79.0, IDQL 75.2). The margin is especially large on the medium-play and large-play tasks (e.g., 90 versus 84.5 for IDQL on medium-play, and 85 versus 63.5 for IDQL on large-play), where accurate OOD rejection is critical for long-horizon navigation.

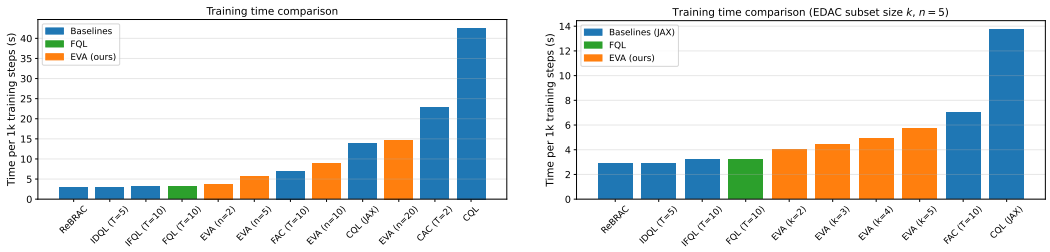
5.1.2 OGBench experiments

Scalable offline RL algorithms are getting more and more attention in the offline RL field. To further show the broad applicability of our method, we combined it with FQL (Park et al., 2025b), a most representative scalable offline RL algorithm, to test its effectiveness on the well-known challenging OGBench tasks. We specifically select the AntSoccer, cube-double, and humanoidmaze-large tasks, which are among the most challenging and algorithm-discriminating tasks in OGBench, where prior methods exhibit the largest performance gaps. We implement our algorithms based on FQL's source code provided by the authors. We add a EDAC loss (An et al., 2021) and EVA loss on FQL's critic while keeping the flow-based actor unchanged. We compare to the recent offline RL methods on OGBench, including FQL (Park et al., 2025b), Q-flow (Doo et al., 2026), and FAC (Chae et al., 2026). All models are trained for 1M steps, and following the FQL evaluation protocol (Park et al., 2025b), the reported score of each seed is the average success rate over its final three evaluations. The results are presented in table 3. The full hyperparameters are reported in Appendix A.1.

On OGBench, FQL-EDAC-EVA sets the best score on all three tasks and raises the average to 47.0, a 72% improvement over FQL (27.3) and a 33% improvement over the strongest baseline FAC (35.3). The gains are largest on the hardest manipulation task, cube-double, where our method reaches 62.3 versus 38 for FAC and 36 for FQL, and on the long-horizon humanoidmaze task (25.3 versus 15 for FAC). Importantly, comparing FQL-EDAC (which already augments FQL with a plain EDAC-style ensemble) against FQL-EDAC-EVA isolates the contribution of our variance-suppression term: adding EVA lifts the score from 48.0 to 53.5 on AntSoccer, from 12.2 to 62.3 on cube-double, and from 7.4 to 25.3 on humanoidmaze. The fact that a plain ensemble helps only modestly on cube-

Table 3: Average success rate (%) of FQL-EDAC-EVA (ours) and baselines on OGBench tasks (default). Each score is the mean over 4 random seeds. The best result per row is bolded.

	Diffusion Policies			Gaussian Policies			Flow Policies						
	Normal (ensemble_size ≤ 2)			Normal (ensemble_size ≤ 2)			Normal (ensemble_size ≤ 2)					Uncertainty-based (ensemble_size ≥ 2)	
Dataset	IDQL	SRPO	CAC	IQL	BC	ReBRAC	FAWAC	FQL	IFQL	FAC	Q-Flow	FQL-EDAC	FQL-EDAC-EVA(ours)
antsoccer-arena-navigate-singletask-v0	0	0	0	3	0	7	12	39	16	53	27	48.0±4.7	**53.5±5.5**
cube-double-play-singletask-task2-v0	16	0	2	1	1	0	2	36	9	38	27	12.2±2.5	**62.3±10.2**
humanoidmaze-large-navigate-singletask-task1-v0	0	0	1	3	0	2	0	7	6	15	14	7.4±5.5	**25.3±5.8**
Avg	5.3	0.0	1.0	2.3	0.3	3.0	4.7	27.3	10.3	35.3	22.7	22.5	**47.0**



(a) Baselines are scaled to the FQL reference, and EVA (ours) is shown for ensemble sizes $n = 2, 5, 10, 20$.

(b) EVA (ours) with ensemble size $n = 5$ and EDAC subset size $k \in \{2, 3, 4, 5\}$.

Figure 3: Wall-clock time of 1,000 training steps in the cube-double environment of OGBench.

double, while the variance-suppressed ensemble yields a five-fold gain, indicates that it is not the ensemble size per se but the re-calibration of ensemble uncertainty that drives the improvement. This is consistent with our central claim that over-dispersed ensemble variance, not its absence, is the bottleneck.

5.2 Training Time

We report the wall-clock time of running 1,000 training steps in the cube-double environment of OGBench (Figure 3a), comparing our method at several ensemble sizes n with representative offline RL baselines scaled to the same FQL reference. The training time of our method grows with the ensemble size n: at $n = 2$ it is only slightly slower than FQL, whereas at $n = 20$ it approaches the cost of CQL (JAX) and CAC, and the traditional PyTorch CQL remains more than an order of magnitude slower than FQL. At $n = 5$ our method already attains strong performance while requiring only a modest time overhead, so all OGBench results reported in this paper use $n = 5$. Overall, our method keeps an acceptable training time compared to previous methods. The extra time mainly comes from the EDAC loss, which computes the pairwise cosine similarity among all ensemble critics, a matrix product with $O(n^2)$ complexity in the ensemble size.

On D4RL MuJoCo, where our method is instantiated on top of EDAC, the added variance-suppression term is essentially free. As shown in Table 4, EVA as a plug-in almost does not change the running time of EDAC, incurring only a marginal overhead on top of the base EDAC critic.

Table 4: Wall-clock time per 1,000 training steps on D4RL MuJoCo

Method	Time (s/1k steps)
EDAC	13.2
EDAC + EVA (ours)	13.5

5.3 Ablation Study

We study the effect of the EVA strength coefficient β on the AntSoccer and cube-double tasks of OGBench (Figure 4), which also reveals why EVA improves performance. EVA helps by increasing the separation between out-of-distribution (OOD) and in-distribution (ID) actions: suppressing the in-distribution ensemble variance lowers the disagreement on dataset actions while keeping the disagreement on OOD actions large, so the actor can more easily distinguish OOD actions from

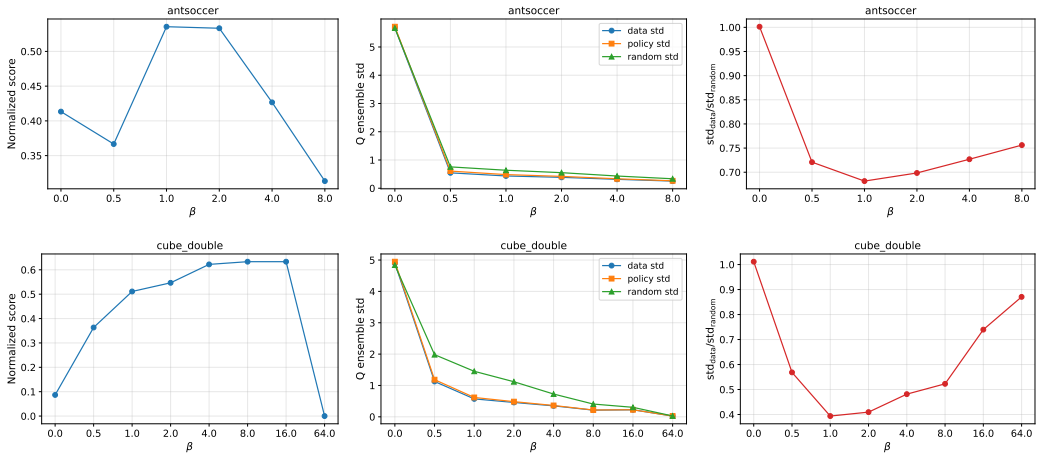


Figure 4: Effect of the EVA strength β on AntSoccer (top) and cube-double (bottom). Left: final normalized score versus β. Middle: the ensemble standard deviation of dataset, policy, and random actions versus β. Right: the ratio of dataset to random-action standard deviation versus β. Each score is the mean over 4 random seeds.

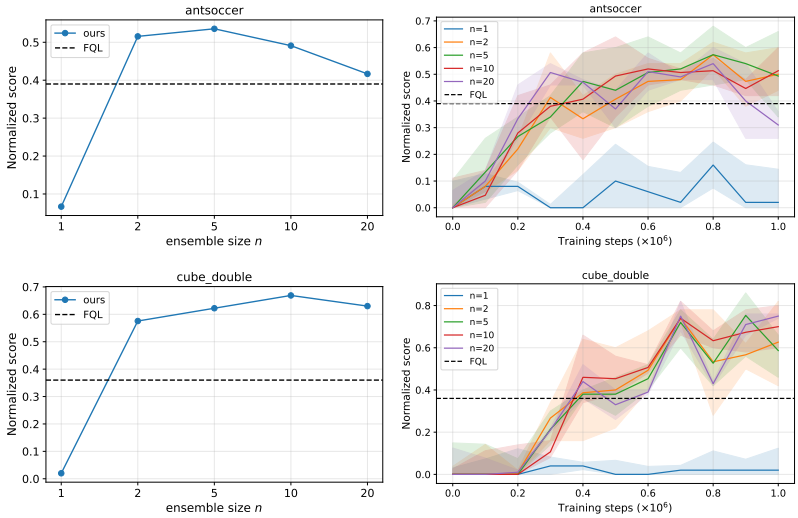


Figure 5: Effect of the ensemble size n on AntSoccer (top) and cube-double (bottom). Left: final normalized score versus n, with the FQL baseline shown as a dashed line. Right: training-process score for each n, shaded by the per-step minimum/maximum. Each score is the mean over 4 random seeds.

in-distribution ones. This separation is quantified by the ratio of the dataset to the random-action ensemble standard deviation (right column of Figure 4). A moderate β is necessary: too-small β fails to widen this gap, while too-large β collapses the ensemble and shrinks the OOD signal as well, both of which degrade the final performance.

We further study the effect of the ensemble size n (Figure 5). A single critic ($n = 1$) performs poorly, but once $n \geq 2$ the performance becomes insensitive to the ensemble size. We therefore use $n = 5$ in all reported experiments.

Finally, we study the sensitivity of EDAC to the subset size k used for the pairwise cosine-similarity penalty (Figure 6). For each $k \in \{2, 3, 4, 5\}$ we record the evaluation success rate at each training step together with its per-step minimum and maximum across seeds; the final score is the mean

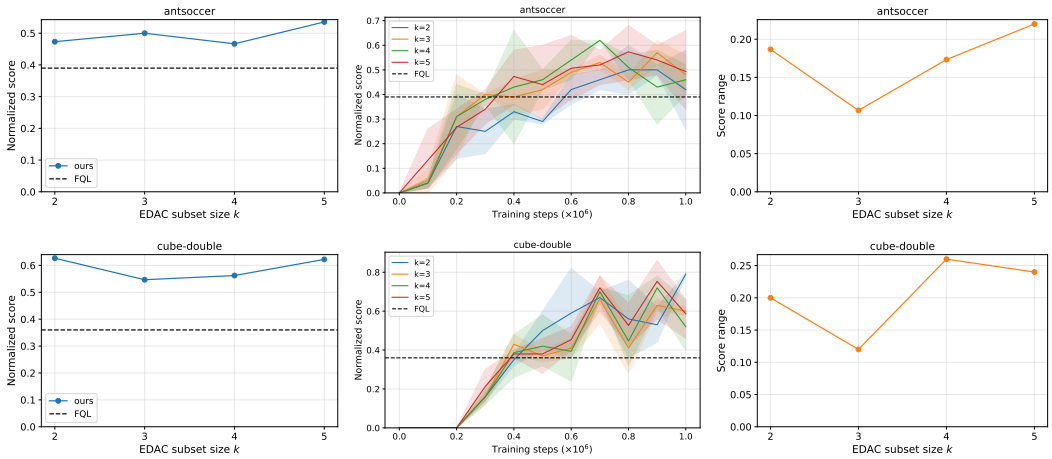


Figure 6: Effect of the EDAC subset size k on AntSoccer (top) and cube-double (bottom). Left: final normalized score versus k, with the FQL baseline shown as a dashed line. Middle: training-process score for each k, shaded by the per-step minimum/maximum. Right: score stability versus k, where stability is the mean of the per-step range over the last three recorded steps. Each score is the mean over 4 random seeds.

over the last three recorded steps, and the stability is the mean of the per-step range (maximum minus minimum) over the same window. The final performance is robust to k: all choices reach a comparable score. The full-ensemble setting $k = n = 5$, which applies the pairwise cosine-similarity penalty over every critic pair, is slightly better overall. When training speed matters, we recommend $k \approx n/2$, which achieves comparable performance and stability while avoiding the extra computation of the full pairwise set.

Reducing k also clearly lowers the training-time overhead (Figure 3b). Because the EDAC pairwise cosine-similarity penalty compares fewer critic pairs as k shrinks, the per-step time grows smoothly with k: at $k = 2$ our method is only marginally slower than FQL, whereas the full $k = 5$ remains faster than CQL (JAX) and FAC. This reinforces the $k \approx n/2$ recommendation above as a practical time/performance trade-off.

6 CONCLUSION

We identified a subtle but critical limitation of ensemble-based offline RL: the variance among ensemble critics is systematically over-dispersed, exceeding the intrinsic action-value variance even on in-distribution data, which corrupts both policy optimization and out-of-distribution detection. To remedy this, we proposed EVA, an in-distribution variance-suppression term that re-aligns ensemble uncertainty with the intrinsic action-value variation. EVA is simple, model-free, and plug-in: by suppressing in-distribution disagreement it widens the gap between in-distribution and out-of-distribution actions, so the actor can more easily tell them apart. Combined with EDAC on D4RL and FQL on OGBench, EVA consistently attains competitive or stronger results with negligible or modest overhead.

Ablations show a clear sweet spot for the suppression strength β—too little leaves the ensemble over-dispersed, while too much collapses it and destroys the OOD signal—insensitivity to the ensemble size n once $n \geq 2$ and to the EDAC subset size k, and that a small $k \approx n/2$ offers a favorable time/performance trade-off.

Extending ensemble-based methods to broader goal-conditioned tasks remains an important direction for future work.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for paper presentation and code completion. We have not used generative AI tools for idea thinking, and experiments data generation are not applicable to this work. Additionally, we used generative AI tools for latex code writing. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

See the ICLR 2027 AI Policy for Authors for more details. This statement should not be more than 1 page.

ETHICS STATEMENT

(This section is **recommended** and does not count toward the page limit.)

If authors feel that their paper submission raises questions regarding the Code of Ethics, they are encouraged to include a paragraph of Ethics Statement (at the end of the main text before references) to address potential concerns where appropriate. Topics include, but are not limited to, studies that involve human subjects, practices to data set releases, potentially harmful insights, methodologies and applications, potential conflicts of interest and sponsorship, discrimination/bias/fairness concerns, privacy and security issues, legal compliance, and research integrity issues (e.g., IRB, documentation, research ethics). This statement should not be more than 1 page.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

It is important that the work published in ICLR is reproducible. Authors are strongly encouraged to include a paragraph-long Reproducibility Statement at the end of the main text (before references) to discuss the efforts that have been made to ensure reproducibility. This paragraph should not itself describe details needed for reproducing the results, but rather reference the parts of the main paper, appendix, and supplemental materials that will help with reproducibility. For example, for novel models or algorithms, a link to an anonymous downloadable source code can be submitted as supplementary materials; for theoretical results, clear explanations of any assumptions and a complete proof of the claims can be included in the appendix; for any datasets used in the experiments, a complete description of the data processing steps can be provided in the supplementary materials. Each of the above are examples of things that can be referenced in the reproducibility statement.

REFERENCES

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

Alex Beeson and Giovanni Montana. Balancing policy constraint and ensemble size in uncertainty-based offline reinforcement learning. *Machine Learning*, 113(1):443–488, 2024.

Jongseong Chae, Jongeui Park, Yongjae Shin, Gyeongmin Kim, Seungyul Han, and Youngchul Sung. Flow actor-critic for offline reinforcement learning. *arXiv preprint arXiv:2602.18015*, 2026.

Huayu Chen, Cheng Lu, Zhengyi Wang, Hang Su, and Jun Zhu. Score regularized policy optimization through diffusion behavior. In *International Conference on Learning Representations*, volume 2024, pp. 10211–10230, 2024.

Zihan Ding and Chi Jin. Consistency models as a rich and efficient policy class for reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 53047–53066, 2024.

JaeHyeok Doo, Byeongguk Jeon, Seonghyeon Ye, Kimin Lee, and Minjoon Seo. Q-flow: Stable and expressive reinforcement learning with flow-based policy. *arXiv preprint arXiv:2605.13435*, 2026.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto and Shixiang Shane Gu. A Minimalist Approach to Offline Reinforcement Learning. *arXiv:2106.06860 [cs, stat]*, December 2021. URL http://arxiv.org/abs/2106.06860. arXiv: 2106.06860.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Kamyar Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters. *Advances in Neural Information Processing Systems*, 35:18267–18281, 2022.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 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.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-Learning for Offline Reinforcement Learning. pp. 13.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

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

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv:2005.01643 [cs, stat]*, November 2020. URL http://arxiv.org/abs/2005.01643. arXiv: 2005.01643.

Jiafei Lyu, Xiaoteng Ma, Xiu Li, and Zongqing Lu. Mildly conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:1711–1724, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Alexander Nikulin, Vladislav Kurenkov, Denis Tarasov, and Sergey Kolesnikov. Anti-exploration by random network distillation. In *International Conference on Machine Learning*, pp. 26228–26244. PMLR, 2023.

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

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *International Conference on Machine Learning (ICML)*, 2025b.

Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *arXiv preprint arXiv:2204.12581*, 2022.

Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. CORL: Research-oriented deep offline reinforcement learning library. In *3rd Offline RL Workshop: Offline RL as a "Launchpad"*, 2022. URL https://openreview.net/forum?id=SyAS49bBcv.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:11592–11620, 2023.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Da Wang, Yi Ma, Ting Guo, Hongyao Tang, Wei Wei, and Jiye Liang. Fans: A flatness-aware network structure for generalization in offline reinforcement learning. *Advances in Neural Information Processing Systems*, 38:149075–149098, 2026.

Kerong Wang, Hanye Zhao, Xufang Luo, Kan Ren, Weinan Zhang, and Dongsheng Li. Bootstrapped transformer for offline reinforcement learning. *arXiv preprint arXiv:2206.08569*, 2022.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Junghyuk Yeom, Yonghyeon Jo, Jungmo Kim, Sanghyeon Lee, and Seungyul Han. Exclusively penalized q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37:113405–113435, 2024.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 17943–17970, 2025.

Yujia Zhang, Lin Li, Wei Wei, Jianguo Wu, Yi Ma, and Jiye Liang. Efficient offline reinforcement learning via peer-influenced constraint. In *International Conference on Learning Representations*, volume 2026, pp. 68202–68232, 2026.

A Appendix

A.1 Hyperparameters

The unified (default) and task-specific hyperparameters used in our experiments are summarized in Table 5, Table 6, and Table 7.

Table 5: Hyperparameters on D4RL MuJoCo.

(a) Unified (default) hyperparameters.

Hyperparameter	Value
`hidden_dim`	$(256, 256, 256)$
`num_critics`	10
γ	0.99
τ	5×10^{-3}
`q_agg`	min (followed EDAC)
`actor_learning_rate`	3×10^{-4}
`critic_learning_rate`	3×10^{-4}
`alpha_learning_rate`	3×10^{-4}
`batch_size`	256
`training_steps`	3M
`normalize_reward`	False
`normalize`	False (True for wk-m & wk-m-r)

(b) Task-specific hyperparameters.

Task	β	η
`hc-m`	0.1	0.1
`hp-m`	0.1	0.1
`wk-m`	0.1	5
`hc-m-r`	0.1	1
`hp-m-r`	0.1	0.1
`wk-m-r`	0.1	1
`hc-m-e`	0.1	1
`hp-m-e`	0.1	1
`wk-m-e`	0.1	5

Table 6: Hyperparameters on D4RL AntMaze.

(a) Unified (default) hyperparameters.

Hyperparameter	Value
`actor-lr`	1×10^{-4}
`critic-lr`	3×10^{-4}
`hidden-dims`	$(256, 256, 256)$
γ	0.99
τ	0.005
`normalize-reward`	False
`num-critics`	10
`q_normalized`	True
`batch-size`	256
`q_agg`	mag
`training_steps`	3M

(b) Task-specific hyperparameters.

Task	η	β	α
`antmaze-medium-play`	0.1	0.01	0.01
`antmaze-medium-diverse`	0.1	0.01	0.01
`antmaze-large-play`	0.1	0.01	0.02
`antmaze-large-diverse`	0.1	0.01	0.02

Table 7: Hyperparameters on OGBench.

(a) Unified (default) hyperparameters.

Hyperparameter	Value
lr	3×10^{-4}
batch_size	256 (512 for humanoid-large)
actor_hidden_dims	$(512, 512, 512, 512)$
value_hidden_dims	$(512, 512, 512, 512)$
num_critics	5
layer_norm	True
actor_layer_norm	False
discount	0.99 (0.995 for antsoccer & humanoid-large)
τ	0.005
q_agg	msg
alpha_decay	0.25
flow_steps	10
normalize_q_loss	False
edac_k	3
training_steps	1M

(b) Task-specific hyperparameters.

Task	η	β	α
antsoccer-arena-navigate-singletask-v0	1	1	10
cube-double-play-singletask-task2-v0	0.1	4	300
humanoidmaze-large-navigate-singletask-task1-v0	0.1	0.1	20