Simple Ingredients for Offline Reinforcement Learning

Edoardo Cetin [1] **Andrea Tirinzoni** [2] **Matteo Pirotta** [2] **Alessandro Lazaric** [2] **Yann Ollivier*** [2] **Ahmed Touati*** [2]

Abstract

Offline reinforcement learning algorithms have proven effective on datasets highly connected to the target downstream task. Yet, by leveraging a novel testbed (MOOD) in which trajectories come from heterogeneous sources, we show that existing methods struggle with diverse data: their performance considerably *deteriorates* as data collected for related but different tasks is simply *added* to the offline buffer. In light of this finding, we conduct a large empirical study where we formulate and test several hypotheses to explain this failure. Surprisingly, we find that targeted scale, more than algorithmic considerations, is the key factor influencing performance. We show that simple methods like AWAC and IQL with increased policy size overcome the paradoxical failure modes from the inclusion of additional data in MOOD, and notably outperform prior state-of-the-art algorithms on the canonical D4RL benchmark.

1. Introduction

Offline reinforcement learning (RL) holds the promise of overcoming the costs and dangers of direct interaction with the environment by training agents exclusively on logged data. However, naively applying off-policy algorithms to this setting has been shown prone to instabilities due to their natural tendency to extrapolate beyond the given data (Fujimoto et al., 2019; Kumar et al., 2019). To address this issue, *policy constrained* methods propose minimal modifications to off-policy actor-critic algorithms aimed at keeping the learned policy close to the data distribution (Levine et al., 2020; Fujimoto & Gu, 2021; Nair et al., 2020; Kostrikov et al., 2022; Garg et al., 2023; Fujimoto et al., 2023). For instance, TD3+Behavior Cloning (TD3+BC, Fujimoto & Gu, 2021) achieves this by regularizing the actor loss with the divergence between the learned policy and the data-generating policy, while Advantage Weighted Actor Critic (AWAC, Nair et al., 2020) seeks a policy maximizing the data likelihood weighted by its exponentiated advantage function. Later extensions of AWAC also modify the critic loss to avoid querying actions outside the given data by learning a value function, e.g., by expectile regression in Implicit Q-learning (IQL, Kostrikov et al., 2022) and Gumbel regression in Extreme Q-learning (XQL, Garg et al., 2023). This class of methods can be easily integrated with online fine-tuning, even leading to several successful applications for real-world tasks (Lu et al., 2022; Nair et al., 2023).

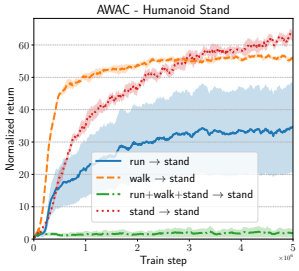

Figure 1. The AWAC algorithm learns to stand when trained on data generated by an agent learning to either stand, walk, or run, but completely fails on the union of these three datasets.

However, current offline RL methods still fail in simple settings. Hong et al. (2023b;c) showed that if the data contains many low-return and few high-return trajectories, policy constrained methods are unnecessarily conservative and fail to learn good behavior. Singh et al. (2023) report a similar effect on heteroskedastic datasets where the variability of behaviors differs across different regions of the state space.

Realistic scenarios often involve data coming from many heterogeneous sources, such as agents trained for different tasks or demonstrations of diverse behaviors (Lu et al., 2022; Wagener et al., 2022). Despite the richness of these data, we show, through a novel testbed (MOOD), that existing offline RL methods can still fail: simply concatenating datasets collected from different tasks significantly and consistently *hurts* performance. Counter-intuitively, this happens even for tasks where *training succeeds on any of the individual subsets*. This is strikingly shown in the example in Fig. 1 with the Humanoid environment: using offline datasets collected for the run, walk, or stand tasks is enough to learn to stand, yet, simply combining them into a single multi-task superset leads AWAC to near-zero performance.

In light of this observation, our contributions are as follows. **1)** We introduce MOOD, a new testbed for offline RL based on the DeepMind Control suite (Tassa et al., 2018)

*Equal contribution [1]Sakana AI, Tokyo, Japan, work done at Meta [2]FAIR at Meta, Paris, France. Correspondence to: Edoardo Cetin <edo@sakana.ai>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

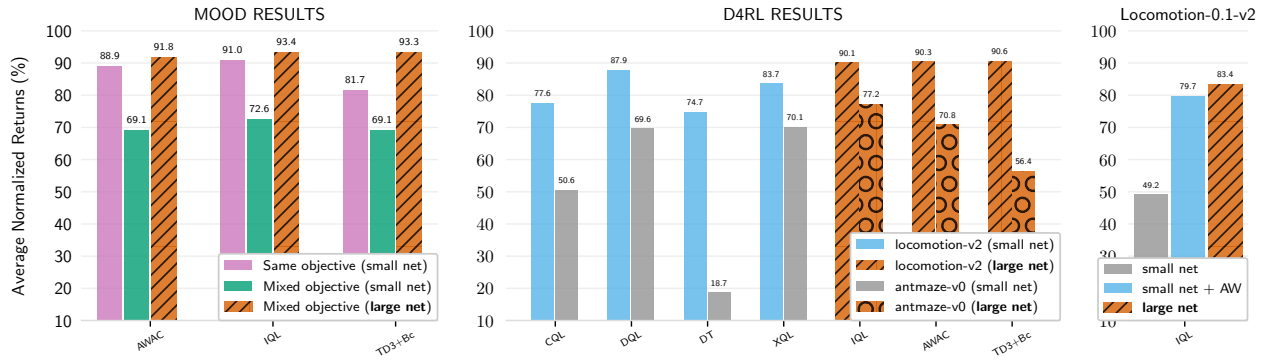

Figure 2. Average performance on the same- and mixed-objective datasets from MOOD (*left*), and the locomotion and antmaze datasets from D4RL (*right*). The large networks are simple MLPs for MOOD and modern architectures (Bjorck et al., 2021) for D4RL, and all involve an ensemble of 5 critics (Sec. 4). "AW" in the last plot denotes the sampling strategy of Hong et al. (2023a) for unbalanced data.

which involves datasets with mixed data from different behaviors. We use it to illustrate the negative impact of data diversity on offline RL methods. **2)** We formulate several hypotheses on the limitations that lead to such a negative result, including over-conservatism of the algorithm, scale, variance, and epistemic uncertainty, while proposing principled solutions to mitigate them. **3)** Through a systematic empirical analysis, we test these hypotheses and solutions across three representative algorithms (TD3+BC, AWAC, IQL) and various hyperparameters, conducting over 50,000 experiments. Surprisingly, we find that scale, specifically of the policy network, emerges as the key factor impacting performance: a simple increase in the number of hidden layers and units in significantly improves the performance of all candidate algorithms on the diverse data in MOOD. **4)** We show similar positive results in the canonical D4RL benchmark, where AWAC and IQL, with increased network sizes, *surpass state-of-the-art performance* on the locomotion and antmaze datasets (Fig. 2), and *match the performance of sophisticated sampling strategies* recently proposed specifically for the unbalanced variants of the locomotion datasets (Hong et al., 2023a).

We provide access to our code at: `https://github.com/facebookresearch/offline_rl`.

2. Preliminaries

Reinforcement learning problems are typically modeled by a Markov Decision Process (MDP, Bellman, 1957), i.e., a tuple $(S, A, P, p_0, r, \gamma)$ with a state space S, an action space A, transition dynamics $P : S \times A \to \text{Prob}(S)$, initial state distribution $p_0 \in \text{Prob}(S)$, reward function $r : S \times A \to \mathbb{R}$, and discount factor $\gamma \in [0, 1)$. The goal of an RL problem is to learn an optimal policy $\pi^\star : S \to \text{Prob}(A)$, which maximizes the expected sum of discounted rewards (a.k.a. the return): $\pi^\star \in \arg\max_\pi \mathbb{E}^\pi[\sum_{t=0}^\infty \gamma^t r(s_t, a_t)]$, where the expectation is under trajectories $\tau = (s_0, a_0, s_t, \dots)$ with $s_0 \sim p_0$, $a_t \sim \pi(s_t)$, and $s_{t+1} \sim P(s_t, a_t)$ for all $t \geq 0$. The algorithms analyzed in this paper are based on the popular off-policy actor-critic framework for continuous control (Silver et al., 2014; Lillicrap et al., 2015). To optimize a parameterized policy π_θ (i.e., the *actor*), these algorithms learn *critic* models to approximate its action-value function $Q^{\pi_\theta}(s, a) := \mathbb{E}^{\pi_\theta}[\sum_{t=0}^\infty \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a]$, the value function $V^{\pi_\theta}(s) := \mathbb{E}_{a\sim\pi_\theta(s)}[Q^{\pi_\theta}(s, a)]$, or the advantage function $A^{\pi_\theta}(s, a) := Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s)$. The algorithms we consider build on top of TD3 (Fujimoto et al., 2018), which models the critic with two randomly-initialized Q-functions (Q_{ϕ_1}, Q_{ϕ_2}). The parameters $\phi \in \{\phi_1, \phi_2\}$ of each Q-function are optimized independently via temporal difference (TD) on a dataset $\mathcal{B}$ as

$$\arg\min_\phi \mathbb{E}_{(s,a,s',r)\sim\mathcal{B}} \left[(Q_\phi(s, a) - y)^2 \right], \tag{1}$$

where $y = r + \gamma \mathbb{E}_{a'\sim\pi_\theta(s')} \left[\min(Q_{\bar{\phi}_1}(s', a'), Q_{\bar{\phi}_2}(s', a')) \right]$ is the TD target and $(\bar{\phi}_1, \bar{\phi}_2)$ are delayed versions of the parameters (ϕ_1, ϕ_2) used to stabilize training. The policy is then optimized in alternation with the Q-functions as

$$\arg\max_\theta \mathbb{E}_{s\sim\mathcal{B}, a\sim\pi_\theta(s)} \left[Q_{\phi_1}(s, a) \right]. \tag{2}$$

These two optimization steps are commonly referred to as *policy evaluation* and *policy improvement*. While in the canonical online RL setting the agent iteratively alternates learning ϕ and θ with collecting data in the environment, in offline RL the buffer $\mathcal{B}$ is collected apriori with some unknown behavior policy $\pi_\mathcal{B}$ and no further interaction with the environment is allowed. In this case, it is well known that applying off-policy algorithms out of the box is prone to instabilities and several modifications have been proposed to counteract their natural tendency to extrapolate beyond the provided data (Kumar et al., 2019; Fujimoto et al., 2019).

2.1. Offline RL algorithms with policy constraints

We describe the offline RL methods employed in our analyses: TD3 + Behavior Cloning (TD3+BC, Fujimoto & Gu, 2021), Advantage Weighted Actor Critic (AWAC, Nair et al., 2020), and Implicit Q-learning (IQL, Kostrikov et al., 2022). We focus on these specific approaches due to their simplicity and popularity: they all build on top of TD3, a state-of-the-art algorithm for off-policy RL, while adding incremental levels of conservatism in its actor and critic components.

TD3+BC. TD3+BC minimally deviates from TD3, by adding a behavioral cloning term to the policy improvement objective of Equation 2:

$$\arg\max_{\theta} \mathbb{E}_{a,s\sim\mathcal{B},a'\sim\pi_\theta(s)} \left[Q_{\phi_1}(s,a') + \alpha\overline{Q} \log \pi_\theta(a|s) \right],$$

where $\overline{Q}$ is the absolute Q-value averaged over each mini-batch and α is a scaling hyper-parameter. Note that maximizing $\log \pi_\theta(a|s)$ with actions from $\mathcal{B}$ corresponds to minimizing the forward KL divergence $D_{\mathrm{KL}}(\pi_{\mathcal{B}}(\cdot|s)||\pi_\theta(\cdot|s))$.

AWAC. Similarly to TD3+BC, AWAC keeps the same critic update (1) as TD3 while modifying the actor update (2) as

$$\arg\max_{\theta} \mathbb{E}_{a,s\sim\mathcal{B}} \left[\frac{\exp(A_{\phi_1}(s,a)/\beta)}{Z} \log \pi_\theta(a|s) \right], \quad (3)$$

where $A_{\phi_1}(s,a) = Q_{\phi_1}(s,a) - \mathbb{E}_{a'\sim\pi_\theta(s)}[Q_{\phi_1}(s,a')]$ and $Z = \mathbb{E}_{s,a\sim\mathcal{B}}\left[\exp(A_{\phi_1}(s,a)/\beta)\right]$, while β is a temperature hyper-parameter. In tabular settings, this is equivalent to to minimizing $D_{\mathrm{KL}}(\pi^\star_{\mathcal{B}}(\cdot|s)||\pi_\theta(\cdot|s))$, where $\pi^\star_{\mathcal{B}}$ is the policy maximizing the advantage $A_{\phi_1}(s,a)$ subject to an inverse KL constraint forcing $D_{\mathrm{KL}}(\pi^\star_{\mathcal{B}}(\cdot|s)||\pi_{\mathcal{B}}(\cdot|s)) \leq \epsilon$ (Peters & Schaal, 2007; Peng et al., 2019).

IQL. IQL keeps the policy improvement (8) of AWAC, but modifies its critic to learn a parametric model of the value function V_ψ using expectile regression:

$$\arg\min_{\psi} \mathbb{E}_{s,a\sim\mathcal{B}}[L_2^\tau(Q(s,a) - V_\psi(s))], \quad (4)$$

where $Q(s,a) := \min(Q_{\bar{\phi}_1}(s,a), Q_{\bar{\phi}_2}(s,a))$, $L_2^\tau(u) = |\tau - 1_{\text{if }(u<0)}|u^2$, and $\tau \in (0,1)$ is a hyper-parameter. It then learns the action-value functions (Q_{ϕ_1}, Q_{ϕ_2}) by modifying the TD targets in (1) as $y = r + \gamma V_\psi(s')$. The main advantage over the critic update of TD3 is that IQL never queries the learned Q-functions on actions outside the dataset.

3. Offline RL with Diverse Data

Prior offline RL methods have been extensively tested and validated using well-known benchmarks such as D4RL (Fu et al., 2020) and RL-unplugged (Gulcehre et al., 2020), but the datasets within these benchmarks exhibit a significant data collection bias towards the specific offline task considered for evaluation. Recent works (Hong et al., 2023b;c) showed that offline methods tend to fail when the dataset is unbalanced (e.g., when most trajectories have low return). Here we provide a complementary analysis to highlight the challenges of incorporating **diverse** data sources. To this end, we introduce Multi Objective Offline DMC (MOOD), a new testbed for offline RL to focus on this relevant problem dimension.

3.1. The MOOD testbed

We build MOOD on top of the DeepMind Control suite (Tassa et al., 2018), spanning four environments (15 total tasks) of increasing complexity (cheetah, walker, quadruped, and humanoid, see Fig. 6 in App. A) with several mixed- or cross-task setups for each. For each environment, we first collect data by training behavior policies for different objectives (see Tab. 7 in App. A), including both traditional reward maximization on DMC tasks (e.g., walk, run, or stand) and the exploration-focused intrinsic motivation from Random Network Distillation (RND, Burda et al., 2019). We train agents for several million steps based on the difficulty of the environment and gather data by randomly sub-sampling 10% of the resulting replay buffers. We then merge the data coming from some subset of tasks, and relabel them for a possibly different target task, thus building several datasets for benchmarking offline RL methods. Each MOOD dataset is denoted as "*domain source-tasks* $\rightarrow$ *target-task*", where "*domain*" is the considered environment (e.g., Walker), "*source-tasks*" lists the objectives whose data was merged, and "*target-task*" is the task used to relabel the rewards (i.e., the task to be solved on this dataset). Depending on the source and target tasks, we obtain different classes of datasets (see App. A for the details):

- *Same-objective datasets* involve a single source task equal to the target task, akin to traditional offline benchmarks (e.g., *Humanoid stand* $\rightarrow$ *stand* in Fig. 1).
- *Cross-objective datasets* involve a single source task which is different from the target task (e.g., *Humanoid walk* $\rightarrow$ *stand* in Fig. 1).
- In *mixed-objective datasets*, the source tasks include all the tasks available in the chosen environment plus optionally RND (e.g., *Walker mixed[+RND]* $\rightarrow$ *walk*).

3.2. The paradoxes of incorporating diverse data

We use MOOD to test our candidate offline RL methods (TD3+BC, AWAC, and IQL) and highlight how they struggle with increasing data diversity. We use a shallow network architecture (2 hidden layers of 256 units with ReLUs in between) for both the actor and the critic of all algorithms, as it is common in existing implementations. For each experiment, (i.e., pair of algorithm and dataset), we perform a grid

Env/Algorithm	IQL	AWAC	TD3	TD3+BC
	Same objective datasets			
cheetah	95.1 ± 1.1	97.2 ± 0.8	58.9 ± 6.6	95.7 ± 1.3
humanoid	74.8 ± 3.1	70.6 ± 3.3	3.5 ± 0.9	47.1 ± 7.7
quadruped	94.5 ± 0.6	91.9 ± 1.7	45.5 ± 3.7	87.3 ± 3.0
walker	95.8 ± 0.9	95.8 ± 0.9	76.5 ± 3.5	96.5 ± 0.6
Total	360.3	355.5	184.4	326.7
	Mixed objective datasets			
cheetah	79.1 ± 5.1	91.1 ± 2.5	81.4 ± 6.3	88.1 ± 4.0
humanoid	26.1 ± 3.2	18.2 ± 4.0	3.9 ± 0.7	11.7 ± 3.2
quadruped	88.7 ± 1.1	74.8 ± 1.9	61.6 ± 3.7	80.8 ± 2.6
walker	90.9 ± 2.3	92.2 ± 2.0	92.9 ± 2.6	95.5 ± 1.6
Total	284.8	276.4	239.8	276.2
Average change	-26.5%	-28.65%	23.09%	-18.26%

Table 1. Average performance (plus/minus standard error) over all tasks per environment on the MOOD datasets. All algorithms use the shallow architecture (2 hidden layers of 256 units) commonly employed in the literature. "Average change" reports the relative performance differences when swapping the same-objective datasets with their mixed-objective MOOD supersets.

search over the hyperparameters specific to each offline algorithm using 5 random seeds, and select the configurations that lead to the highest cumulative return after 1.5×10^6 optimization steps (5×10^6 for humanoid). We report the performance of each algorithm as the average cumulative return normalized by the highest return of a trajectory present in the dataset. We show in Table 1 the results for the same- and mixed-objective datasets[1] averaged over all tasks in each environment. See App. C for all the details and results.

Offline RL struggles with increased data diversity. When examining the ability of offline RL methods to leverage auxiliary data from different tasks on the mixed-objective datasets, we consistently observe a counter-intuitive phenomenon: *adding* data from various sources significantly reduces the performance of all the considered offline RL algorithms. Note that no subsampling occurs when merging the task-specific datasets: the mixed-objective dataset is a superset of the same-objective dataset on which the algorithms work seamlessly. This phenomenon seems more pronounced in harder tasks, with a performance drop higher than 50% for all offline RL algorithms in Humanoid.

TD3 benefits from increased data diversity. On the contrary, the performance of plain TD3, an algorithm originally designed for online RL, improves significantly with more diverse data. This is not surprising, as it is known that "exploratory data", such as the one generated by RND, allows non-conservative algorithms like TD3 to counteract extrapolation tendency and achieve higher performance in many considered tasks (Yarats et al., 2022). It is thus natural to ask why offline RL algorithms incur the opposite behavior.

[1]All mixed-objective datasets used in the main paper contain RND data except for humanoid, where the complexity of the domain makes RND generate useless samples.

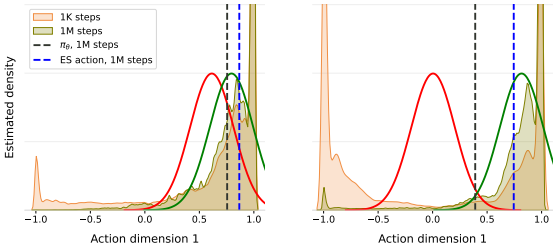

Figure 3. Optimal advantage-weighted distribution $\pi^\star_{\mathcal{B}}$ (shaded areas) and its Gaussian projection (solid curves) after 1K and 1M optimization steps of AWAC on cheetah run with the same-objective (*left*) and the mixed-objective (*right*) datasets. Dashed lines indicate the actions chosen during evaluation using either the mean of the learned policy (black) or ES (blue) after 1M steps. Distributions are plotted for a randomly-chosen state and action dimension.

4. On the Failure of Existing Algorithms

We list several hypotheses on why existing offline RL methods struggle with the mixed-objective data in MOOD. For each of them, we propose simple remedies that can be seamlessly integrated without altering the nature of the method itself. We empirically test each of these hypotheses in Sec. 5.

Hypothesis 1: over-conservatism

The first candidate hypothesis is over-conservatism: all the considered methods force the learned policy to stay close to the data distribution. This is clearly beneficial when the data contains mostly high-return trajectories for the desired task (e.g., in D4RL or MOOD same-objective datasets). Still, it can have a detrimental effect when this condition fails to hold (e.g., in MOOD mixed-objective datasets). In fact, as the data contains behaviors far from the desired one (e.g., a humanoid running or walking when the task is to stand), the learned policy may be forced to put probability mass over poor actions, hence drifting from optimality. This phenomenon was observed in recent works on unbalanced datasets containing mostly low-return trajectories (Hong et al., 2023b;c). Other works also observed that, while constraining or regularizing the policy stabilizes training, it may degrade the evaluation performance in particular cases (Kumar et al., 2019; Singh et al., 2023; Yu et al., 2023).

This conservatism may be amplified in the considered algorithms, which all fit *Gaussian* policies regularized by a *forward KL* term to the data distribution. Hence, given that the data distribution is often multi-modal (e.g., in the mixed-objective datasets) and the mean-seeking tendency of the forward KL divergence, the learned policy's mean is unlikely to reflect the apex of the underlying target distribution

Algorithm 1 Online deployment with evaluation sampling

Input: actor π_θ, critic Q_ϕ, number of action samples M
Get initial state: $s \sim p_0$
while not done **do**
  Sample actions: $a_1, \dots a_M \sim \pi_\theta(\cdot|s)$
  $a^\star \leftarrow \arg\max_{a \in \{a_1, \dots, a_M\}} Q_\phi(s, a)$
  Play $a^\star$, get state $s \sim P(s, a^\star)$ and reward $r(s, a^\star)$
end while

of behaviors. We illustrate this point for AWAC in Fig. 3, where we plot the target advantage-weighted distribution $\pi^\star_{\mathcal{B}}(a|s) \propto \exp(A_\phi(s,a)/\beta)\pi_{\mathcal{B}}(a|s)$ (estimated through an auto-regressive density model) together with the *Gaussian* policy π that minimizes $D_{\text{KL}}(\pi^\star_{\mathcal{B}}(\cdot|s)||\pi(\cdot|s))$ (i.e., the policy we hope the algorithm to learn). The plots clearly show that, on the mixed-objective dataset, the learned policy's mean is very far from the target distribution's mean[2].

Evaluation sampling (ES). If over-conservatism is really an issue, we propose to address it entirely at *test time* by sampling M actions from the learned policy and selecting the one with the highest Q-value, thus performing a non-parametric step of unconstrained policy improvement to skew the action distribution towards higher performance. We call this approach *evaluation sampling* (ES). See Alg. 1 and Fig. 3. In contrast to similar approaches (Wang et al., 2020; Ghasemipour et al., 2021), ES does not alter training at all, thus preserving the desired policy support constraints. We expect ES to yield positive signal mostly when the whole policy distribution is within the data support (i.e., when the policy is really over-conservative as hypothesized), as otherwise extrapolating beyond it may hinder performance.

Hypothesis 2: model scale

As the mixed-objective datasets are a strict superset of their same-objective counterparts, both the actor and critic networks are required to model wider regions of the state-action space. This may lead the networks to spend capacity in modeling unnecessary quantities, as some of these regions may actually be useless for the task at hand (e.g., it is not strictly necessary to model the action values of states corresponding to the humanoid running when learning how to stand). Thus, with networks of limited capacity, one may expect a loss of accuracy in regions that are actually important for learning the given task. We thus hypothesize that network scale may be one of the factors leading to the performance drop on mixed-objective data. In our experiments, we first test this hypothesis with wider and deeper variants of the network architectures commonly employed in the literature, without other changes.

Large modern architectures. Some recent works showed that merely adopting very deep architectures can be prone to instabilities in RL (Andrychowicz et al., 2020; Bjorck et al., 2021; Ota et al., 2021). To make sure we do not run into this issue, we also test an alternative *modern architecture* proposed by Bjorck et al. (2021). Such an architecture was shown to enable stable training thanks to a combination of the fully-connected residual blocks commonly used in transformer models (Vaswani et al., 2017) with spectral normalization (Miyato et al., 2018) (illustrated Fig. 7, App. B). This architecture was also shown to help counteract the plasticity loss, specifically of the critic network (Cetin & Celiktutan, 2023), a phenomenon that frequently occurs when training with non-stationary data and objectives (Ash & Adams, 2020; Dohare et al., 2023) as in RL (Lyle et al., 2023; Nikishin et al., 2022; D'Oro et al., 2023; Schwarzer et al., 2023). Hence, we examine if such a non-stationarity may be significantly affecting training with mixed-objective data also in the offline setting (cf. the drift of the target policy $\pi^\star_{\mathcal{B}}$ in Fig. 3).

Hypothesis 3: epistemic uncertainty/overestimation bias

It is well-known that off-policy actor-critic algorithms like DDPG and TD3 are subject to the same Q-value overestimation bias as in Q-learning with discrete actions (Fujimoto et al., 2018). In online RL, some level of overestimation may be acceptable (e.g., to encourage exploration), and the learner could always correct wrong estimates by gathering further data from the environment. But in offline RL it is important to guarantee pessimistic Q-value estimates for state-action pairs not sufficiently covered by the dataset (Jin et al., 2021). Special care needs to be taken for algorithms, like TD3+BC and AWAC, that query the learned value functions on actions outside the given dataset: these algorithms may result in erroneous overestimations without the possibility to ever correct them, hence yielding poor evaluation performance. This phenomenon may be further exacerbated in mixed-objective data, where errors can propagate due to over-generalization across different regions of the state-action space. We thus hypothesize that the solution proposed in TD3, consisting of training an ensemble of two independent Q-functions with TD targets involving the minimum between them (Eq. 1), may not be sufficient to counteract this issue. Some works (Lan et al., 2020; Chen et al., 2021b) indeed showed that using a larger ensemble of critics can reduce both the Q-value estimation bias and variance. We thus test this approach in our experiments by training an ensemble of n randomly-initialized Q-functions $(Q_{\phi_1}, \dots, Q_{\phi_n})$ while independently optimizing their parameters via TD learning as in (1). Following Cetin & Celiktutan (2023), we redefine the TD targets in (1)

[2]While the fact that Gaussian policies poorly fit the target distribution was initially another hypothesis behind over-conservatism, we discarded it as we found such policies to have a useful regularization effect, and the usage of more expressive distributions or reverse (mode-seeking) KL led to performance collapse.

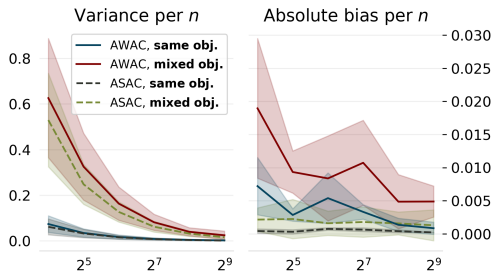

Figure 4. Empirical bias and variance of the AWAC's and ASAC's objective estimators for different batch sizes n on cheetah run. Each point is averaged over 1000 randomly sampled minibatches of size n at equally-spaced checkpoints saved during training.

as $y = r + \gamma \mathbb{E}_{a' \sim \pi_\theta(s')} \left[\overline{Q}(s, a) \right]$, where

$$\overline{Q}(s,a) := \frac{1}{n} \sum_i Q_{\phi_i}(s,a) - \frac{\lambda}{n^2 - n} \sum_{i,j} |Q_{\phi_i}(s,a) - Q_{\phi_j}(s,a)|,$$

with a hyperparameter $\lambda \geq 0$. This gives us more flexibility in the aggregation of the ensemble estimates: for $n = 2$ and $\lambda = 0.5$, we recover the same update rule as TD3, while for other choices of λ we can control the level of pessimism.

Hypothesis 4: bias and variance of advantage weighting

The final hypothesis focuses on advantage-weighted algorithms (AWAC and IQL) using the policy improvement objective from Eq. 8, which is implemented by employing a weighted importance sampling (WIS) estimator:

$$J^{\mathrm{AW}}(\theta) = \mathbb{E}_{(a_{1:n}, s_{1:n}) \sim \mathcal{B}} \left[\sum_{i=1}^{n} w(s_i, a_i) \log \pi_\theta(a_i | s_i) \right], \quad (5)$$

where the weights $w(s_i, a_i) = \frac{\exp(A_\phi(s_i, a_i)/\beta)}{\sum_{j=1}^{n} \exp(A_\phi(s_j, a_j)/\beta)}$ are normalized over a minibatch of size n. It is known that this estimator is both biased and introduces higher variance than directly sampling from the desired target distribution (Hesterberg, 1995). On mixed-objective data, as the distributions involved become more complex, it is natural to expect bias and variance to increase (cf. Fig. 4). We thus hypothesize this fact to be one of the reasons behind the performance drop of the considered advantage-weighted algorithms.

If this is really the case, we propose a very simple workaround: we can directly and tractably sample from the desired target distribution by avoiding altogether the need for weights in the objective. Formally, we define a modified sampling data distribution for policy improvement:

$$\mathcal{B}^\star(s,a) := \frac{1}{Z} \mathcal{B}(s,a) \exp(A_\phi(s,a)/\beta) \quad (6)$$

where $\mathcal{B}(s,a)$ denotes the distribution obtained by i.i.d. sampling from the buffer, while $Z = \sum_{s,a \in \mathcal{B}} \exp(A_\phi(s,a)/\beta)$.

Algorithm 2 Advantage Sampled Actor Critic (ASAC)

Input: offline data $\mathcal{B}$
while not done **do**
  // Actor update
  Sample batch $b_{\mathrm{ac}} = \{(s,a)\}$ from $\mathcal{B}^\star$ (see Eq. 6)
  Take gradient step on θ to maximize (7) on b_{ac}
  // Critic update (TD3)
  Sample batch $b_{\mathrm{cr}} = \{(s,a,s',r)\}$ from $\mathcal{B}$
  Take gradient step on ϕ to minimize (1) on b_{cr}
  // Update sum-tree for $\mathcal{B}^\star$
  $\mathcal{B}^\star(s,a) \leftarrow \exp(A_\phi(s,a)/\beta)$ for all $(s,a) \in b_{\mathrm{cr}} \cup b_{\mathrm{ac}}$
end while

To sample from $\mathcal{B}^\star$ efficiently without explicitly computing Z, we design a *logsumexp-tree* inspired by the sum-tree data structure used for prioritized sampling (Schaul et al., 2015), where we store the scaled advantages $A_\phi(s,a)/\beta$ and work entirely in log-space. Then, we simply train π to maximize the likelihood of the data sampled from $\mathcal{B}^\star$:

$$\hat{J}^{\mathrm{AS}}(\theta) = \mathbb{E}_{(s,a) \sim \mathcal{B}^*} [\log \pi_\theta(a|s)]. \quad (7)$$

It is easy to see that this estimator, which aims at directly projecting π_θ onto $\pi^\star_{\mathcal{B}}$, is *unbiased* for the AWAC objective, i.e., its expectation is equal to the objective in (8).

We call the resulting approach *advantage sampled actor critic* (ASAC, see Alg. 2). [3] Note that it bears some similarities with methods that alter the data distribution via weighting techniques (Hong et al., 2023b;c; Yue et al., 2023). The main difference is that these methods compute weights for offline trajectories only once at the start of training based on the returns/advantages of the behavior policy, while ASAC's sampling distribution adaptively evolves over time.

5. Empirical Results

We systematically evaluate three candidate algorithms (TD3+BC, AWAC, and IQL) combined with the algorithmic design considerations from Sec. 4: the use of a large simple MLP network, a large modern architecture, an ensemble of critics, evaluation sampling, and advantage sampling.

In this Section, we report the key results needed for our claims, while referring the reader to App. C for the complete evaluation. Our main findings indicate that network capacity, specifically for the policy, is the main factor affecting mixed-task performance. First, we provide the results from our final implementations and comparisons, where for the large architectures we employ critics with 3 hidden layers of 256 units each and actors with 5 hidden layers of 1024 units each. For the modern architecture, we use critics with a single modern block (cf. Fig. 7) with 256 as hidden

[3] Alg. 2 only updates $\mathcal{B}^\star(s,a)$ on each minibatch rather than the whole buffer, so some stored values of A_ϕ may grow "stale": this re-introduces some bias for (7). Fig. 4 shows this bias is limited.

Env/Algorithm	IQL		ASAC		AWAC		TD3		TD3+BC	
Architecture (large)	Modern	Simple	Modern	Simple	Modern	Simple	Modern	Simple	Modern	Simple
Total (2 critics, no ES)	369.7	373.0	347.4	331.3	359.5	355.9	228.9	202.5	358.9	370.2
Total (5 critics, no ES)	370.0	373.6	364.9	355.9	367.5	367.3	228.9	207.6	365.8	373.1
	5 critics / large simple architecture									
ES	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓
cheetah	92.5 ± 1.8	89.4 ± 3.4	94.9 ± 1.1	95.4 ± 1.4	95.7 ± 1.0	95.1 ± 1.5	81.0 ± 6.8	85.1 ± 5.4	94.5 ± 1.7	94.6 ± 1.9
humanoid	92.1 ± 1.6	80.9 ± 2.0	91.7 ± 1.0	66.2 ± 0.5	88.3 ± 1.2	78.6 ± 3.4	4.7 ± 1.1	2.9 ± 0.9	86.2 ± 1.3	77.3 ± 3.6
quadruped	94.0 ± 0.7	97.1 ± 0.3	74.7 ± 1.3	89.9 ± 1.1	86.7 ± 2.0	95.6 ± 0.7	26.9 ± 2.7	31.7 ± 3.6	94.9 ± 0.6	98.1 ± 0.5
walker	95.0 ± 1.3	95.7 ± 1.3	94.6 ± 1.5	97.4 ± 0.8	96.6 ± 1.1	98.2 ± 0.6	95.1 ± 1.9	95.3 ± 1.8	97.5 ± 0.3	99.0 ± 0.4
Total	373.6	363.2	355.9	348.9	367.3	367.4	207.6	215.0	373.1	368.9
Total (max over ES)	378.6		372.7		378.3		218.3		378.1	
	5 critics / large modern architecture									
ES	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓
cheetah	93.2 ± 1.6	89.0 ± 3.7	94.7 ± 1.3	94.2 ± 1.2	95.0 ± 1.3	93.8 ± 1.9	76.3 ± 5.5	81.6 ± 6.6	91.9 ± 1.9	91.2 ± 3.2
humanoid	89.9 ± 1.8	78.2 ± 2.8	90.4 ± 0.8	79.1 ± 2.2	91.4 ± 0.6	78.5 ± 3.7	3.3 ± 0.4	3.0 ± 0.5	83.5 ± 1.4	76.4 ± 3.9
quadruped	92.1 ± 0.8	96.7 ± 0.3	83.2 ± 1.5	92.2 ± 0.8	85.4 ± 2.0	92.2 ± 1.5	56.1 ± 2.0	57.9 ± 3.0	92.1 ± 0.9	93.8 ± 0.9
walker	94.6 ± 1.4	95.3 ± 1.3	96.5 ± 0.5	96.7 ± 0.7	95.8 ± 1.1	95.5 ± 1.5	93.1 ± 1.9	94.8 ± 2.1	98.2 ± 0.4	98.1 ± 0.5
Total	370.0	359.2	364.9	362.2	367.5	360.0	228.9	237.4	365.8	359.4
Total (max over ES)	376.3		377.6		377.2		242.0		368.1	

Table 2. Average performance computed over all the tasks per each MOOD environment on the mixed-objective datasets, with totals denoting their sum over environments, using an ensemble of 5 critics and (optionally) ES with $M = 50$ samples. The top and bottom tables report results with the *large* 'simple' and *large* 'modern' actor architectures, respectively. We highlighting improved/worsened results from the default algorithms with shallow actor architectures in Table 1

agent	IQL		ASAC		AWAC		TD3+BC	
ES	✗	✓	✗	✓	✗	✓	✗	✓
Antmaze-v0 (5 critics)	65.2±3.5	77.2±2.8	64.5±4.1	72.6±3.8	64.7±4.1	70.8±4.0	50.7±5.9	51.9±7.0
Antmaze-v0 (10 critics)	63.4±3.3	76.0±2.8	65.3±4.0	72.6±3.6	69.8±4.4	72.6±4.0	49.8±7.1	44.2±7.6
Locomotion-v2 (5 critics)	88.1±3.4	72.3±4.3	84.1±3.2	80.7±3.5	87.0±3.3	81.6±3.3	82.6±2.9	84.9±2.5
Locomotion-0.1-v2 (10 critics)	83.1±4.2	50.6±4.6	64.1±4.0	48.6±4.0	54.6±4.5	51.7±4.2	49.6±4.5	45.5±3.8

Table 3. Average performance across all datasets in each category of D4RL. All algorithms use the large modern architecture.

dimension and actors with 2 modern blocks with 1024 as hidden dimension. This makes the simple and modern networks of equal capacity, disregarding the marginal increase in parameters introduced by layer normalization. Then, we provide some additional experiments that investigate and analyze specific reasons behind our practical observations on scaling.

For each design choice and task, we perform a hyperparameter sweep over the learning rate, the temperature β (for AWAC and IQL), the regularization strength α (for TD3+BC), and the expectile τ (for IQL). We then report the cumulative return of the best configurations averaged over 5 random seeds.

5.1. MOOD evaluation

Table 2 reports the results on the mixed-objective datasets in MOOD. We notably observe that all algorithms bridge the performance gap with same-objective datasets by using the larger architecture (cf Tab. 1), while all other conjectured solutions either marginally help or do not help at all on top of it. In Appendix C, we also quantitatively examine the other hypotheses in isolation, showing how they are all generally insufficient. Specifically, we note the following: i) The modern architecture does not appear to provide any advantage over the simple one ii) ASAC's performance is on par with AWAC, indicating that the variance of the AWAC estimator is not a limiting factor iii) Increasing the number of critics appears to yield a consistent, though marginal, performance improvement, but at the cost of added compute. This makes us conclude that, among our hypotheses, insufficient policy scale is the key factor affecting the performance drop with mixed objective data.

Finally, to a consistently lesser extent than scale, we also note that ES contributes to performance improvements, particularly in Quadruped, but it has a detrimental effect on Humanoid. This discrepancy may be attributed to the availability of RND data with good coverage for the non-humanoid tasks, which makes it easier for conservative methods to skew the whole policy distribution within the support of the data, hence enabling ES to safely improve performance.

This is not the case on the Humanoid datasets comprising only "purposeful" trajectories, where extrapolation has actually a detrimental effect. Further evidence for this conjecture is given in App. C.5, where we show that ES consistently improves performance on pure RND data from ExoRL (Yarats et al., 2022).

5.2. D4RL evaluation

Table 4 reports the results of IQL, AWAC, TD3+BC, and ASAC with large architectures, 5 critics, and ES (in most of the settings). We also report previously-published results for state-of-the-art algorithms: decision transformer (DT, Chen et al., 2021a), conservative Q-learning (CQL, Kumar et al., 2020), extreme Q-learning (XQL, Garg et al., 2023), and Diffusion Q-learning (DQL, Wang et al., 2023). Overall, IQL, AWAC, and ASAC with the large modern architecture surpass the state of the art, while the performance of TD3+BC still falls behind in the antmaze tasks. We emphasize that, while CQL and XQL use smaller architectures, the purpose of these experiments is not to establish which of our algorithms is best, but rather to showcase that several existing simple strategies with increased policy scale consistently outperform state-of-the-art more complex approaches. We further note that the performance gap between large and small networks is particularly pronounced for antmaze. This aligns with the observations from MOOD: D4RL locomotion data is collected with a protocol similar to same-objective MOOD, while the Antmaze datasets, being generated by multi-goal reaching policies, involve increased data diversity.

Unbalanced data. Hong et al. (2023a) observed that offline RL algorithms struggle on the unbalanced locomotion datasets consisting of 90% trajectories generated by the uniform policy and 10% of expert or medium trajectories. They conjectured over-conservatism being the main issue, as the performance of policy-constrained methods is tightly coupled to the (poor) behavior policy. As a solution, they suggested sampling trajectories proportionally to their cumulative return, essentially rebalancing and artificially skewing the distribution trajectories back towards higher performance. Our results (Tab. 4) suggest this ad-hoc strategy might be superflous: simply using larger networks, while training with standard uniform sampling, achieves generally comparable and even some better results.

The role of ES. As shown in Tab. 3, ES improves the performance on antmaze by a large margin, while actually hurting in all locomotion datasets. This is likely to be due to a similar phenomenon as in the mixed-objective MOOD data: good data coverage from antmaze enables accurate learning of the Q-functions within the policy support, as opposed to the narrower distributions of the locomotion tasks where extrapolation is more error-prone.

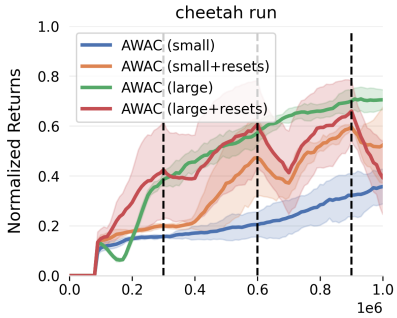

Figure 5. The AWAC algorithm's performance on the mixed objective cheetah run offline task, with different policy scales and periodic policy network resets.

5.3. Why policy scale?

	AWAC configuration					
Critic	Large		Large		Small	
Policy	Small		Large		Large	
ES	✗	✓	✗	✓	✗	✓
cheetah	86.82 ± 3.49	94.46 ± 2.25	95.29 ± 1.04	93.83 ± 2.39	95.7 ± 1.0	95.1 ± 1.5
humanoid	26.43 ± 3.59	41.18 ± 3.18	78.21 ± 3.81	76.39 ± 3.74	88.3 ± 1.3	78.6 ± 3.4
quadruped	96.31 ± 1.7	99.58 ± 0.75	95.12 ± 0.92	98.55 ± 0.43	86.7 ± 2.0	95.6 ± 0.7
walker	89.4 ± 2.52	98.72 ± 0.25	95.51 ± 1.16	97.41 ± 0.97	96.6 ± 1.1	98.2 ± 0.6
Total	298.96	333.94	364.13	366.18	**367.3**	**367.4**

Table 5. Average performance over all tasks per environment on the MOOD mixed objective datasets, with AWAC and different scales for the policy and critic networks.

In this subsection, we focus on the AWAC algorithms and perform more targeted experiments motivating and analyzing our particular scaling strategy. First, in Table 5, we show how scaling the critic network on the mixed objective offline MOOD datasets is alone insufficient, and seems to even slightly hinder overall performance as compared to only scaling the actor. We additionally note that scaling the critic tends to be also disproportionally more computationally expensive. We believe the importance of actor scale is to counteract early saturation (i.e., plasticity loss) specifically in the actor network. In fact, as shown in Figures 3 and 9 (App. E), in the offline setting the actor optimization is very non-stationary (the optimal targets considerably differ between 1K and 1M steps) and small actors seem to struggle to approximate the optimal distribution later on in training.

We note our findings and analysis puts offline RL in direct contrast to online RL where plasticity loss seems a predominant issue for the critic network (Lyle et al., 2023; Nikishin et al., 2022; D'Oro et al., 2023).

To provide further evidence to validate this hypothesis, we introduced a new experiment where we analyze the effects from periodic resets to the actor network. We note this was introduced by Nikishin et al. (2022) to mitigate plasticity loss in the critic for online RL, and is now a key practice adapted and performed in several sample-efficient algorithms (D'Oro et al., 2023; Schwarzer et al., 2023). In our case, we instead focus on the actor network and re-initialize

Dataset/Algorithm	DT	CQL	XQL	DQL	IQL		ASAC		AWAC		TD3+BC	
Architecture	orig.	orig.	orig.	orig.	modern	simple	modern	simple	modern	simple	modern	simple
halfcheetah-medium-expert-v2	86.8	91.6	94.2	96.8	93.4±0.1	93.5±0.2	94.5±0.2	88.5±1.2	94.3±0.4	**100.3±1.0**	92.7±0.2	94.1±0.8
halfcheetah-medium-replay-v2	36.6	45.5	45.2	47.8	49.9±0.1	52.0±0.2	54.5±0.2	54.9±0.3	54.4±0.3	57.5±0.1	57.9±0.1	**59.1±0.6**
halfcheetah-medium-v2	42.6	44.0	48.3	51.1	58.2±0.1	63.0±0.1	61.3±0.5	61.1±0.2	61.4±0.1	64.6±0.2	64.6±0.3	**70.8±0.4**
hopper-medium-expert-v2	107.6	105.4	111.2	111.1	110.5±0.2	110.0±0.2	110.9±0.2	109.8±0.2	110.3±0.1	110.4±0.2	**111.8±0.3**	109.6±0.4
hopper-medium-replay-v2	82.7	95.0	100.7	101.3	101.1±0.5	94.7±1.9	101.4±0.4	95.2±1.4	101.9±0.6	**102.5±0.5**	100.6±0.2	98.7±2.1
hopper-medium-v2	67.6	58.5	74.2	90.5	**98.1±0.8**	96.4±2.0	94.6±4.7	75.6±1.6	96.4±2.8	81.7±16.2	94.9±0.8	93.2±0.7
walker2d-medium-expert-v2	108.1	108.8	112.7	109.6	112.9±0.3	114.1±0.4	112.1±0.2	113.5±0.3	112.8±0.1	**115.5±0.3**	**115.0±0.4**	**115.5±0.9**
walker2d-medium-replay-v2	66.6	77.2	82.2	95.5	**96.2±0.2**	94.2±0.7	93.9±0.5	91.1±1.7	94.7±0.4	**96.3±1.0**	90.5±0.3	87.7±0.4
walker2d-medium-v2	74.0	72.5	84.2	87.0	**90.4±0.2**	**89.8±0.4**	86.1±0.1	84.6±0.2	87.1±0.1	87.0±0.3	87.1±0.2	87.2±0.3
Locomotion-v2 total	672.6	698.5	752.9	790.7	810.7	807.7	809.3	774.3	813.1	**815.7**	815.0	816.0
antmaze-large-diverse-v0	0.0	14.9	49.0	56.6	60.4±1.2	**64.4±1.3**	49.9±1.5	49.2±0.8	43.0±3.4	32.0±1.1	18.8±1.5	8.7±1.3
antmaze-large-play-v0	0.0	15.8	46.5	46.4	**54.2±1.1**	26.2±2.3	43.8±2.4	12.3±1.0	42.2±0.8	17.3±0.7	8.3±5.4	5.6±3.5
antmaze-medium-diverse-v0	0.0	53.7	73.6	78.6	82.9±1.1	85.6±0.6	82.3±0.8	**89.1±1.5**	83.2±1.5	85.8±1.1	75.0±7.5	61.9±14.1
antmaze-medium-play-v0	0.0	61.2	76.0	76.6	82.9±0.7	**86.5±0.7**	**84.9±1.4**	**86.7±1.2**	**85.6±1.7**	84.6±1.0	70.5±1.7	24.2±10.0
antmaze-umaze-diverse-v0	53.0	84.0	82.0	66.2	**86.6±0.7**	83.9±0.9	81.1±2.9	72.1±1.5	72.8±3.3	48.5±10.3	71.7±4.9	69.4±3.7
antmaze-umaze-v0	59.2	74.0	93.8	93.4	96.1±0.7	96.6±0.4	98.7±0.1	**99.0±0.2**	97.8±0.5	98.2±0.3	94.1±1.2	96.0±1.2
Antmaze-v0 total	112.2	303.6	420.9	417.8	**463.1**	443.2	440.8	408.5	424.8	366.3	338.4	265.9

Dataset/Algorithm	CQL	IQL	TD3+BC	IQL		ASAC		AWAC		TD3+BC	
Architecture	orig.	orig.	orig.	modern	simple	modern	simple	modern	simple	modern	simple
halfcheetah-random-expert-0.1-v2	45.8	**91.3**	78.4	65.8±2.2	76.3±2.1	56.2±6.6	61.5±6.7	66.7±3.0	78.3±1.2	18.3±1.2	30.7±1.2
halfcheetah-random-medium-0.1-v2	45.8	43.1	47.8	48.9±0.3	52.7±0.2	53.2±0.4	53.6±4.1	54.6±0.1	58.9±0.2	57.5±0.3	**60.7±0.7**
hopper-random-expert-0.1-v2	109.7	**111.5**	107.7	109.0±0.4	95.4±2.2	73.4±5.3	41.8±2.7	32.3±4.8	78.4±1.5	80.2±4.4	77.0±2.0
hopper-random-medium-0.1-v2	66.6	57.1	56.4	**93.9±2.3**	70.5±3.0	69.0±2.1	66.7±2.3	71.7±1.2	76.3±3.8	71.1±1.9	60.5±1.7
walker2d-random-expert-0.1-v2	108.1	109.3	**110.1**	107.8±0.3	99.2±0.6	99.5±2.0	101.1±0.6	93.6±3.1	61.3±15.5	41.9±17.0	11.6±0.8
walker2d-random-medium-0.1-v2	66.3	65.8	74.2	75.0±0.5	73.3±0.6	65.9±4.7	75.0±0.9	47.0±6.9	**78.4±3.1**	57.5±4.3	53.1±10.4
Locomotion-0.1-v2 total	442.3	478.1	474.6	**500.5**	467.4	417.2	399.8	365.9	431.5	326.5	293.5

Table 4. Performance on the locomotion-v2 and antmaze-v0 datasets from the D4RL benchmark (*top*), and on the unbalanced variants of the locomotion-v2 datasets (*bottom*). For our candidate algorithms we use *large* architectures, 5 critics (10 for the unbalanced data), and report the best result between using or not ES (with $M = 50$ samples). The second block of the bottom table reports the performance of the algorithms tested by Hong et al. (2023a) with their modified sampling distribution (values taken from their Github repository).

all weights every 300K steps, each time performing 50K subsequent 'warmup' policy improvement steps keeping the critic fixed (not counted when reporting performance curves). Our results in Figure 5, illustrate that our reset strategy visibly improves the final performance of standard AWAC with small actors, even while facing visible periodic instabilities after each reset. Yet, the same strategy does not seem to benefit our scaled implementation, validating the effectiveness of the increased policy capacity to counteract early saturation.

6. Conclusions

We provided an empirical analysis for some of the key difficulties of current offline RL methods, highlighting the unexpected consequences of training from mixtures of diverse data sources. In our study, targeted scaling of the policy appears more important than algorithmic considerations in overcoming these and improving performance: two simple algorithms (AWAC, IQL) surpass state-of-the-art methods on the standard D4RL benchmark, and even TD3+BC comes reasonably close. In contrast, further algorithmic refinements yield only limited benefits. This questions the common trend of designing increasingly more complicated algorithms, as even simple approaches seem sufficient with proper implementations. We believe that understanding and analyzing what really matters is key for demystifying the field, and hope our work will serve as a valuable resource to empower future advancements.

Impact Statement

This paper presents work whose goal is to advance the field of offline RL. Given the nature of our contribution, its societal implications are bound to the broad potential implications of advancing autonomous agents. In this regard, poor regulation and misuse of such advancements may accentuate inequalities and cause harm. However, we believe these concerns to be offset by the field's current potential in tackling some of society's most relevant problems.

References

Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., et al. What matters in on-policy reinforcement learning? a large-scale empirical study. *arXiv preprint arXiv:2006.05990*, 2020.

Ash, J. T. and Adams, R. P. On warm-starting neural network training. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

Bellman, R. A Markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957. ISSN 0022-2518.

Bjorck, J., Gomes, C. P., and Weinberger, K. Q. Towards deeper deep reinforcement learning. *arXiv preprint arXiv:2106.01151*, 2021.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1lJJnR5Ym.

Cetin, E. and Celiktutan, O. Learning pessimism for reinforcement learning. In *The 37th AAAI Conference on Artificial Intelligence (AAAI-23)*. AAAI Press, 2023.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34, 2021a.

Chen, X., Wang, C., Zhou, Z., and Ross, K. W. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=AY8zfZm0tDd.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Dohare, S., Hernandez-Garcia, J. F., Rahman, P., Sutton, R. S., and Mahmood, A. R. Loss of plasticity in deep continual learning, 2023.

D'Oro, P., Schwarzer, M., Nikishin, E., Bacon, P.-L., Bellemare, M. G., and Courville, A. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OpC-9aBBVJe.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Fujimoto, S., van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *ICML*, pp. 1582–1591, 2018. URL http://proceedings.mlr.press/v80/fujimoto18a.html.

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

Fujimoto, S., Chang, W.-D., Smith, E. J., Gu, S. S., Precup, D., and Meger, D. For sale: State-action representation learning for deep reinforcement learning. *arXiv preprint arXiv:2306.02451*, 2023.

Garg, D., Hejna, J., Geist, M., and Ermon, S. Extreme q-learning: Maxent RL without entropy. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SJ0Lde3tRL.

Germain, M., Gregor, K., Murray, I., and Larochelle, H. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pp. 881–889. PMLR, 2015.

Ghasemipour, S. K. S., Schuurmans, D., and Gu, S. S. Emaq: Expected-max q-learning operator for simple yet effective offline and online rl. In *International Conference on Machine Learning*, pp. 3682–3691. PMLR, 2021.

Gulcehre, C., Wang, Z., Novikov, A., Paine, T., Gómez, S., Zolna, K., Agarwal, R., Merel, J. S., Mankowitz, D. J., Paduraru, C., et al. Rl unplugged: A suite of benchmarks for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:7248–7259, 2020.

Hesterberg, T. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2): 185–194, 1995.

Hong, Z., Agrawal, P., des Combes, R. T., and Laroche, R. Harnessing mixed offline reinforcement learning datasets via trajectory weighting. In *ICLR*. OpenReview.net, 2023a.

Hong, Z., Kumar, A., Karnik, S., Bhandwaldar, A., Srivastava, A., Pajarinen, J., Laroche, R., Gupta, A., and Agrawal, P. Beyond uniform sampling: Offline reinforcement learning with imbalanced datasets. *CoRR*, abs/2310.04413, 2023b.

Hong, Z.-W., Agrawal, P., des Combes, R. T., and Laroche, R. Harnessing mixed offline reinforcement learning datasets via trajectory weighting. In *The Eleventh International Conference on Learning Representations*, 2023c. URL https://openreview.net/forum?id=OhUAblg27z.

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pp. 5084–5096. PMLR, 2021.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

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

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*. 2020.

Lan, Q., Pan, Y., Fyshe, A., and White, M. Maxmin q-learning: Controlling the estimation bias of q-learning. In *International Conference on Learning Representations*, 2020.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Lu, Y., Hausman, K., Chebotar, Y., Yan, M., Jang, E., Herzog, A., Xiao, T., Irpan, A., Khansari, M., Kalashnikov, D., et al. Aw-opt: Learning robotic skills with imitation and reinforcement at scale. In *Conference on Robot Learning*, pp. 1078–1088. PMLR, 2022.

Lyle, C., Zheng, Z., Nikishin, E., Pires, B. A., Pascanu, R., and Dabney, W. Understanding plasticity in neural networks, 2023.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Nair, A., Zhu, B., Narayanan, G., Solowjow, E., and Levine, S. Learning on the job: Self-rewarding offline-to-online finetuning for industrial insertion of novel connectors from vision. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7154–7161. IEEE, 2023.

Nikishin, E., Schwarzer, M., D'Oro, P., Bacon, P.-L., and Courville, A. The primacy bias in deep reinforcement learning. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16828–16847. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/nikishin22a.html.

Ota, K., Jha, D. K., and Kanezaki, A. Training larger networks for deep reinforcement learning. *arXiv preprint arXiv:2102.07920*, 2021.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Peters, J. and Schaal, S. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pp. 745–750, 2007.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Schwarzer, M., Ceron, J. S. O., Courville, A., Bellemare, M. G., Agarwal, R., and Castro, P. S. Bigger, better, faster: Human-level atari with human-level efficiency. In *International Conference on Machine Learning*, pp. 30365–30380. PMLR, 2023.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. 2014.

Singh, A., Kumar, A., Vuong, Q., Chebotar, Y., and Levine, S. Reds: Offline rl with heteroskedastic datasets via support constraints. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Tarasov, D., Kurenkov, V., Nikulin, A., and Kolesnikov, S. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. d. L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wagener, N., Kolobov, A., Vieira Frujeri, F., Loynd, R., Cheng, C.-A., and Hausknecht, M. Mocapact: A multi-task dataset for simulated humanoid control. *Advances in Neural Information Processing Systems*, 35:35418–35431, 2022.

Wang, Z., Novikov, A., Zolna, K., Merel, J. S., Springenberg, J. T., Reed, S. E., Shahriari, B., Siegel, N., Gulcehre, C., Heess, N., and de Freitas, N. Critic regularized regression. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7768–7778. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/588cb956d6bbe67078f29f8de420a13d-Paper.pdf.

Wang, Z., Hunt, J. J., and Zhou, M. Diffusion policies as an expressive policy class for offline reinforcement learning. In *ICLR*. OpenReview.net, 2023.

Yarats, D., Brandfonbrener, D., Liu, H., Laskin, M., Abbeel, P., Lazaric, A., and Pinto, L. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *arXiv preprint arXiv:2201.13425*, 2022.

Yu, L., Yu, T., Song, J., Neiswanger, W., and Ermon, S. Offline imitation learning with suboptimal demonstrations via relaxed distribution matching. In *AAAI*, pp. 11016–11024. AAAI Press, 2023.

Yue, Y., Kang, B., Ma, X., Huang, G., Song, S., and Yan, S. Offline prioritized experience replay. *arXiv preprint arXiv:2306.05412*, 2023.

Appendix

A. MOOD details

Figure 6. Continuous control environments in MOOD.

As introduced in Section 3, Multi Objective Offline DMC (MOOD) allows to evaluate offline agents for tasks with increasing levels of complexity, and assesses their ability to make use of additional data coming from behavior policies trained with different objectives. Our benchmark is based on four environments (see Fig. 6), fifteen tasks, and eighteen base datasets. Each base dataset is collected with a unique objective. We consider either traditional reward maximization objectives for one of the DMC tasks we use for offline training, or the intrinsic objective from Random Network Distillation (RND) (Burda et al., 2019). The only exception is for the humanoid environment, where we only consider the reward maximization objectives. The reason for our choice is that the humanoid model appears quite unstable, with the agent easily losing its balance and collapsing to the ground, a situation from which recovering appears very difficult. Mainly due to this challenge, we found that optimizing an agent with an RND objective fails to capture almost any meaningful kind of behavior, in stark contrast to the other considered environments.

To obtain each of the base datasets for the considered DMC tasks, we start by collecting replay buffer data from training five TD3 agents using different random seeds for a varying number of steps based on the difficulty of the relative environments. Then, we merge the different whole replay buffers, which we note contain trajectories collected throughout the full training procedure, as we do not impose any size limitation during training. We provide the hyper-parameters of the employed TD3 agents in Table A.

Hence, as described in Section 3, we then proceed to relabel all the rewards in each base dataset for all other tasks based on the same environment to produce the same, cross, and mixed objective datasets. When evaluating agents on any MOOD dataset, the reported results represent a normalized percentage, which we simply computed by dividing the collected returns with task-specific targets based on the highest trajectory returns in each of the relative datasets. We refer to Table 7 for the task-specific details regarding dataset collection and offline training.

Online TD3 hyper-parameters	
buffer size $\|B\|$	∞
batch size $\|b\|$	512
minimum data to train	5000
optimizer	Adam
learning rate	0.0003
policy delay	2
discount γ	0.99
polyak coefficient ρ	0.995
policy/Q network hidden layers	2
policy/Q network hidden dimensionality	256
exploration noise	0.2

Table 6. Agent hyper-parameters used for data collection in MOOD.

Environment	$\|S\|$	$\|A\|$	Base dataset objective	Collection steps	Subsampled size	Normalization return target
cheetah	17	6	walk	5×1M	500K	990
			run	5×1M	500K	800
			walk_backward	5×1M	500K	990
			run_backward	5×1M	500K	550
			RND	5×2M	1M	N/A
walker	24	6	walk	5×1M	500K	970
			run	5×1M	500K	730
			stand	5×1M	500K	990
			spin	5×1M	500K	990
			RND	5×2M	1M	N/A
quadruped	78	12	walk	5×1M	500K	940
			run	5×1M	500K	800
			stand	5×1M	500K	970
			jump	5×1M	500K	870
			RND	5×2M	1M	N/A
humanoid	67	21	walk	5×10M	5M	900
			run	5×10M	5M	400
			stand	5×10M	5M	960

Table 7. MOOD datasets collection details.

B. Implementation Details

In this section we provide additional information about our experiments.

B.1. Network Architecture

In table 8, we report the activation functions, number of hidden layers, hidden dimension and number of modern blocks we used for small and large networks for each of actor, critic and value networks. Note that the simple and modern networks have equal capacity, disregarding the marginal increase in parameters introduced by layer normalization. Figure 7 portrays a block of the modern architecture.

Components	Simple-Small	Simple-Large	Modern
	activation=ReLu	activation=ReLu	activation=ReLu
Actor	hidden layers = 2 with hidden dim=256	hidden layers = 5 with hidden dim=1024	blocks = 2, with hidden dim=1024
Critic	hidden layers = 2 with hidden dim=256	hidden layers = 3 with hidden dim=256	blocks = 1, with hidden dim=256
Value (only IQL)	hidden layers = 2 with hidden dim=256	hidden layers = 2 with hidden dim=1024	simple hidden layers = 2 with hidden dim=1024

Table 8. Network specification for small, large and modern networks employed by different algorithms

B.2. Algorithmic details

Advantage clipping: Similarly to what done in previous papers we may use advantage clipping in the actor update to avoid numerical overflow due the exponentiation. The actor update is then

$$\arg\max_{\theta} \mathbb{E}_{a,s\sim\mathcal{B}}\left[\frac{\exp(\min\{A_{\phi_1}(s,a), A_{\max}\}/\beta)}{Z}\log\pi_\theta(a|s)\right]. \tag{8}$$

where $Z = \mathbb{E}_{s,a\sim\mathcal{B}}\left[\exp(\min\{A_{\phi_1}(s,a), A_{\max}\}/\beta)\right]$.

Pessimism penalty: As explained in section 4 for the ensemble of critics, we employ a hyperparameter λ to regulate the extent to which we penalize discrepancies among the critics. We maintain a constant value of $\lambda = 0.5$ across all settings,

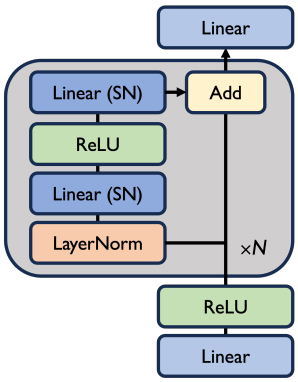

Figure 7. The modern architecture proposed by Bjorck et al. (2021) to enable stable training of very deep networks. Each of N residual blocks (gray part) consists of a layer normalization followed by two linear layers regularized via spectral normalization (SN) with a ReLU non-linearity in between.

Testbed	Alg. Parameters
Common	batch size = 512 discount γ = 0.99 polyak coefficient ρ = 0.995 M = 50
mood	$\lambda = 0.5$, $A_{\max} = \infty$ train steps (cheetah, walker, hopper) = 1.5M train steps (humanoid) = 5M
antmaze-v0	$\lambda = 0$, $A_{\max} = 1$ train steps = 2M
locomotion-0.1-v2	$\lambda = 0.5$, $A_{\max} = \infty$ train steps = 2M
locomotion-v2	$\lambda = 0.5$, $A_{\max} = \infty$ train steps = 2M

Figure 8. The parameters common across algorithms, used for each testbed

except in antmaze, where we observe that $\lambda = 0$ (no penalty) is advantageous. This is likely due to the sparsity of rewards in antmaze domains.

Table 8 highlights the distinct hyperparameters per each testbed.

B.3. Hyperparameter Sweep

Table 9 summarizes the range of hyperparameters sweep we used in our experiments for each algorithms and testbed.

Testbed/Algorithm	IQL	AWAC	ASAC	TD3+BC
mood	$\beta = \{0.1, 0.5, 1.5, 3\}$			$\alpha = \{0, 0.01, 0.1, 1, 10\}$
	$\tau = \{0.7, 0.9\}$			
	learning rate $= \{0.0001, 0.0003\}$			
antmaze-v0	$\beta = \{0.03, 0.06, 0.1, 0.3, 0.5\}$			$\alpha = \{0.01, 0.1, 1, 10\}$
	$\tau = \{0.7, 0.9\}$			
	learning rate $= \{0.0001, 0.0003\}$			
locomotion-0.1-v2	$\beta = \{0.1, 0.5, 1.5, 3, 10, 20\}$		X	$\alpha = \{0.01, 0.1, 1, 10\}$
	$\tau = \{0.7, 0.9\}$			
	learning rate $= \{0.0001, 0.0003\}$			
locomotion-v2	$\beta = \{0.1, 0.5, 1.5, 3\}$	$\beta = \{0.1, 0.3, 0.5, 1.5, 3\}$		$\alpha = \{0, 0.01, 0.1, 1, 10\}$
	$\tau = \{0.7, 0.9\}$			
	learning rate $= \{0.0001, 0.0003\}$			

Table 9. Hyperparameter sweep range per algorithms and per testbed.

C. Full Empirical Results

In this Section, we report our full quantitative granular results, in 'raw' format, listing all relevant hyperparameters for reproducibility with the shared codebase. We provide extensive ablations, carefully validating the impact of 2 vs 5 vs 10 critics, ES vs no ES, and ASAC vs IQL vs AWAC vs TD3+BC across different architectures (simple small, simple large, modern small, modern large). These were collected from thousands of GPU hours, across the D4RL (Locomotion and Antmaze), MOOD, and ExoRL benchmarks, further validating how mixing all other components visibly underperforms across domains as compared to our top baselines. We also provide some further extensions of our algorithms based on concurrent work to improve simple algorithms (Tarasov et al., 2024).

C.1. MOOD: Full Results

	mean/stdErr			
agent	IQL	AWAC	TD3	TD3+BC
actor dim	256	256	256	256
value archi dim	256	0	0	0
num cri	2	2	2	2
critic num layers	2	2	2	2
actor num layers	2	2	2	2
critic model	simple	simple	simple	simple
actor model	simple	simple	simple	simple
value archi model	simple	not avail	not avail	not avail
D	same obj. data	same obj. data	same obj. data	same obj. data
num es	1	1	1	1
task				
cheetah	382.4	388.9	235.5	382.7
cheetah run	89.2 / 0.5	91.6 / 0.3	27.1 / 4.2	86.3 / 1.0
cheetah run backward	93.8 / 0.3	97.6 / 0.1	52.6 / 16.4	96.9 / 0.6
cheetah walk	99.8 / 0.1	100.0 / 0.0	66.4 / 4.0	99.8 / 0.0
cheetah walk backward	99.7 / 0.0	99.6 / 0.0	89.5 / 4.5	99.7 / 0.0
humanoid	229.1	211.8	10.5	141.4
humanoid run	72.0 / 1.3	62.1 / 1.5	2.0 / 0.7	49.3 / 1.6
humanoid stand	67.0 / 1.3	61.7 / 1.1	3.4 / 2.1	10.8 / 1.3
humanoid walk	90.2 / 0.6	88.0 / 0.8	5.1 / 1.4	81.3 / 0.5
quadruped	384.2	367.7	181.9	349.3
quadruped jump	94.6 / 0.5	91.4 / 4.7	41.5 / 7.2	77.8 / 9.4
quadruped run	96.8 / 0.7	88.7 / 5.0	47.3 / 8.9	92.3 / 3.8
quadruped stand	96.6 / 0.3	96.4 / 0.6	60.2 / 3.5	92.3 / 1.5
quadruped walk	96.1 / 0.4	91.2 / 1.1	32.9 / 4.2	86.9 / 5.8
walker	383.4	383.2	306.2	386.1
walker run	89.3 / 0.6	88.7 / 0.3	59.5 / 5.5	93.3 / 1.8
walker spin	98.1 / 0.2	98.5 / 0.2	96.9 / 0.5	98.1 / 0.5
walker stand	98.2 / 0.1	98.1 / 0.2	72.1 / 4.1	97.2 / 0.3
walker walk	97.8 / 0.1	97.8 / 0.3	77.7 / 1.9	97.5 / 0.1

Table 10. Same Objective Dataset: best scores with small architecture

	mean/stdErr			
agent	IQL	AWAC	TD3	TD3+BC
actor dim	256	256	256	256
value archi dim	256	0	0	0
num cri	2	2	2	2
critic num layers	2	2	2	2
actor num layers	2	2	2	2
critic model	simple	simple	simple	simple
actor model	simple	simple	simple	simple
value archi model	simple	not avail	not avail	not avail
load replay buffer	mixed obj. data	mixed obj. data	mixed obj. data	mixed obj. data
num es	1	1	1	1
task				
cheetah	318.8	364.6	331.3	352.6
cheetah run	47.6 / 4.6	73.1 / 1.5	40.7 / 4.8	59.3 / 5.0
cheetah run backward	75.4 / 2.8	93.3 / 0.5	91.8 / 0.5	95.9 / 0.3
cheetah walk	96.3 / 1.2	98.7 / 0.3	99.8 / 0.1	97.7 / 1.1
cheetah walk backward	99.5 / 0.1	99.5 / 0.0	98.9 / 0.6	99.6 / 0.0
humanoid	84.3	54.5	11.6	35.2
humanoid run	20.6 / 0.8	15.4 / 0.4	1.5 / 0.4	7.3 / 1.1
humanoid stand	21.4 / 1.3	1.8 / 1.3	5.6 / 1.3	7.0 / 1.2
humanoid walk	42.3 / 2.2	37.3 / 2.0	4.5 / 0.9	21.0 / 8.5
quadruped	364.9	299.3	246.4	323.4
quadruped jump	91.7 / 1.5	75.5 / 3.7	60.3 / 5.3	77.9 / 1.8
quadruped run	93.9 / 0.8	72.0 / 4.1	54.1 / 2.1	78.7 / 5.4
quadruped stand	90.9 / 0.7	70.2 / 1.7	83.3 / 5.4	87.3 / 3.3
quadruped walk	88.4 / 1.5	81.6 / 4.5	48.7 / 4.2	79.5 / 8.3
walker	365.8	368.9	371.7	383.4
walker run	74.5 / 0.9	78.0 / 3.1	73.8 / 1.8	86.1 / 2.8
walker spin	97.4 / 0.1	96.4 / 0.6	99.3 / 0.1	98.9 / 0.1
walker stand	97.3 / 0.1	97.2 / 0.3	99.4 / 0.1	99.0 / 0.1
walker walk	96.6 / 0.3	97.3 / 0.3	99.3 / 0.3	99.5 / 0.2

Table 11. Mixed Objective Dataset: best scores with small architecture

	mean/stdErr				
agent	IQL	AWAC	TD3	TD3+BC	
actor dim	1024	1024	1024	1024	1024
value archi dim	1024	0	0	0	0
num cri	2	2	2	2	2
critic num layers	3	3	3	3	3
actor num layers	5	5	5	5	5
critic model	simple	simple	simple	simple	simple
actor model	simple	simple	simple	simple	simple
value archi model	simple	not avail	not avail	not avail	not avail
score processing	softmax	asac buffer	softmax	none	none
load replay buffer	mixed obj. data	mixed obj. data	mixed obj. data	mixed obj. data	mixed obj. data
num es	1	1	1	1	1
task					
cheetah	92.6	92.6	95.5	84.5	95.2
cheetah run	80.5 / 0.4	87.3 / 1.1	88.4 / 0.6	45.2 / 9.4	84.0 / 1.7
cheetah run backward	91.1 / 0.5	83.8 / 0.8	94.2 / 0.2	95.6 / 0.7	98.1 / 0.1
cheetah walk	99.3 / 0.5	99.8 / 0.0	99.9 / 0.0	99.4 / 0.5	99.2 / 0.1
cheetah walk backward	99.7 / 0.0	99.5 / 0.1	99.6 / 0.0	97.9 / 1.6	99.5 / 0.1
humanoid	92.1	88.1	85.6	3.0	83.7
humanoid run	98.6 / 0.3	84.7 / 0.3	82.9 / 0.4	1.8 / 0.3	75.1 / 0.9
humanoid stand	84.3 / 0.5	86.2 / 0.5	80.0 / 0.5	5.1 / 1.4	85.5 / 0.7
humanoid walk	93.5 / 0.1	93.5 / 0.2	93.9 / 0.2	2.0 / 1.5	90.6 / 0.2
quadruped	93.9	62.0	78.6	20.4	94.1
quadruped jump	92.0 / 0.5	54.8 / 2.2	76.3 / 5.2	18.8 / 1.1	91.9 / 1.0
quadruped run	92.9 / 1.1	60.5 / 3.1	81.9 / 5.6	15.9 / 1.8	92.7 / 1.0
quadruped stand	94.7 / 0.2	67.1 / 1.0	76.5 / 5.3	28.6 / 2.4	95.5 / 1.1
quadruped walk	96.3 / 0.5	65.6 / 1.5	79.8 / 6.2	18.5 / 5.7	96.4 / 1.0
walker	94.2	88.6	96.1	92.5	97.4
walker run	83.7 / 1.3	63.9 / 0.9	87.2 / 3.0	72.2 / 1.7	95.2 / 0.3
walker spin	98.3 / 0.2	95.8 / 0.2	99.0 / 0.1	99.1 / 0.1	98.1 / 0.1
walker stand	98.0 / 0.1	97.1 / 0.2	98.8 / 0.0	99.5 / 0.1	97.8 / 0.0
walker walk	96.9 / 0.1	97.5 / 0.2	99.3 / 0.1	99.3 / 0.2	98.4 / 0.1
total	373.0	331.3	355.8	200.5	370.4

Table 12. Mixed Objective Dataset: best scores with large architecture and 2 critics

	mean/stdErr				
agent	IQL	AWAC	TD3	TD3+BC	
actor dim	1024	1024	1024	1024	1024
value archi dim	1024	0	0	0	0
num cri	2	2	2	2	2
critic num layers	2	2	2	2	2
actor num layers	2	2	2	2	2
critic model	modern	modern	modern	modern	modern
actor model	modern	modern	modern	modern	modern
value archi model	simple	not avail	not avail	not avail	not avail
score processing	softmax	asac buffer	softmax	none	none
load replay buffer	mixed obj. data	mixed obj. data	mixed obj. data	mixed obj. data	mixed obj. data
num es	1	1	1	1	1
task					
cheetah	92.9	93.6	95.0	80.6	92.7
cheetah run	81.3 / 1.0	83.9 / 0.6	86.7 / 0.5	32.9 / 2.0	73.8 / 1.3
cheetah run backward	90.7 / 0.8	91.2 / 0.5	93.9 / 0.3	90.0 / 1.1	97.3 / 0.2
cheetah walk	99.9 / 0.0	99.8 / 0.0	99.9 / 0.0	99.9 / 0.0	99.9 / 0.0
cheetah walk backward	99.7 / 0.0	99.5 / 0.0	99.6 / 0.0	99.7 / 0.0	99.7 / 0.0
humanoid	91.4	89.5	90.8	2.3	83.2
humanoid run	97.9 / 0.3	87.9 / 0.5	89.3 / 0.9	1.5 / 0.3	77.4 / 0.9
humanoid stand	83.1 / 0.4	86.6 / 0.9	89.0 / 0.5	2.5 / 0.5	82.5 / 0.3
humanoid walk	93.2 / 0.1	94.0 / 0.1	94.1 / 0.2	2.9 / 0.8	89.8 / 0.4
quadruped	91.0	76.8	77.5	52.8	85.2
quadruped jump	87.9 / 0.7	77.6 / 1.0	63.7 / 10.9	56.6 / 4.4	73.7 / 8.5
quadruped run	94.0 / 0.7	77.0 / 5.7	86.8 / 3.9	50.3 / 1.9	88.9 / 2.6
quadruped stand	92.8 / 0.6	79.5 / 1.6	79.0 / 1.8	65.8 / 4.1	90.7 / 3.3
quadruped walk	89.2 / 1.0	73.0 / 1.5	80.4 / 6.6	38.3 / 2.1	87.5 / 5.5
walker	94.4	87.6	96.2	93.3	97.8
walker run	84.1 / 0.5	65.1 / 1.4	90.3 / 0.7	75.4 / 3.5	93.9 / 0.4
walker spin	98.1 / 0.1	95.7 / 0.7	97.3 / 0.3	99.3 / 0.1	98.7 / 0.3
walker stand	97.6 / 0.2	93.8 / 1.4	98.4 / 0.1	99.5 / 0.1	99.1 / 0.1
walker walk	97.9 / 0.2	95.7 / 0.5	98.8 / 0.1	99.0 / 0.3	99.7 / 0.0
total	369.7	347.4	359.5	229.0	358.9

Table 13. Mixed Objective Dataset: best scores with modern architecture and 2 critics

					mean/stdErr					
agent	IQL		ASAC		AWAC		TD3		TD3+BC	
actor dim	1024		1024		1024		1024		1024	
value archi dim	1024		0		0		0		0	
num cri	5		5		5		5		5	
critic num layers	3		3		3		3		3	
actor num layers	5		5		5		5		5	
critic model	simple		simple		simple		simple		simple	
actor model	simple		simple		simple		simple		simple	
value archi model	simple		not avail		not avail		not avail		not avail	
score processing	softmax		asac buffer		softmax		none		none	
load replay buffer	mixed obj. data		mixed obj. data		mixed obj. data		mixed obj. data		mixed obj. data	
num es	1	50	1	50	1	50	1	50	1	50
task										
cheetah	92.5	89.4	94.9	95.4	95.7	95.3	81.0	85.1	94.5	94.6
cheetah run	81.0 / 0.3	64.6 / 2.2	87.9 / 0.7	85.2 / 0.8	89.5 / 0.4	85.0 / 1.2	29.4 / 0.9	45.2 / 5.1	81.6 / 1.2	80.4 / 1.0
cheetah run backward	89.4 / 0.4	94.1 / 0.4	92.5 / 0.2	96.8 / 0.3	94.0 / 0.4	96.5 / 0.1	96.4 / 0.4	96.2 / 0.7	98.2 / 0.1	98.3 / 0.3
cheetah walk	99.8 / 0.1	99.2 / 0.1	99.8 / 0.0	99.9 / 0.0	99.9 / 0.0	100.0 / 0.0	98.6 / 0.7	100.0 / 0.0	98.9 / 0.4	99.9 / 0.1
cheetah walk backward	99.7 / 0.0	99.7 / 0.0	99.1 / 0.1	99.7 / 0.0	99.6 / 0.0	99.7 / 0.0	99.4 / 0.4	99.0 / 0.8	99.4 / 0.0	99.8 / 0.0
humanoid	92.1	80.9	89.2	80.6	88.3	78.6	4.7	2.9	86.2	77.3
humanoid run	98.6 / 0.4	71.9 / 0.5	84.3 / 0.8	65.4 / 1.0	85.9 / 0.9	61.9 / 0.7	2.5 / 0.7	1.8 / 0.4	79.8 / 0.6	58.9 / 0.5
humanoid stand	84.3 / 0.5	81.0 / 0.5	88.5 / 0.4	83.7 / 0.9	84.8 / 0.8	81.4 / 0.6	6.7 / 2.7	4.5 / 1.2	87.3 / 0.4	82.6 / 1.0
humanoid walk	93.4 / 0.3	89.8 / 0.2	94.6 / 0.2	92.5 / 0.2	94.1 / 0.1	92.4 / 0.2	4.9 / 1.9	2.4 / 2.2	91.4 / 0.2	90.3 / 0.2
quadruped	94.0	97.1	74.7	89.9	86.7	95.6	26.9	31.7	94.9	98.0
quadruped jump	91.8 / 0.7	96.8 / 0.3	75.9 / 1.9	92.4 / 0.8	81.6 / 4.0	94.6 / 0.7	27.0 / 4.7	28.2 / 2.8	91.5 / 0.9	97.5 / 0.4
quadruped run	90.9 / 0.8	97.1 / 0.4	70.9 / 1.8	88.3 / 3.2	85.9 / 6.0	93.6 / 2.5	23.5 / 5.3	31.6 / 7.6	95.0 / 0.7	97.9 / 1.6
quadruped stand	96.2 / 0.2	97.1 / 0.7	74.6 / 3.4	91.1 / 2.0	89.5 / 1.2	96.7 / 0.5	33.4 / 4.8	28.4 / 5.4	96.6 / 0.3	99.4 / 0.4
quadruped walk	97.1 / 0.3	97.6 / 0.6	77.6 / 2.8	88.0 / 2.6	89.9 / 3.8	97.7 / 0.4	23.6 / 6.8	38.7 / 11.8	96.5 / 0.6	97.3 / 1.1
walker	95.0	95.7	94.6	97.4	96.6	98.2	95.1	95.3	97.5	99.0
walker run	85.1 / 1.1	86.3 / 1.3	84.7 / 3.0	91.7 / 0.3	89.3 / 2.5	94.0 / 0.3	81.7 / 2.6	82.6 / 2.6	95.3 / 0.3	96.5 / 0.3
walker spin	98.4 / 0.1	98.9 / 0.1	97.7 / 0.5	98.8 / 0.6	99.1 / 0.0	99.4 / 0.0	99.2 / 0.1	99.0 / 0.2	98.4 / 0.1	99.3 / 0.0
walker stand	98.7 / 0.0	98.9 / 0.0	98.4 / 0.3	99.4 / 0.1	98.9 / 0.1	99.6 / 0.0	99.7 / 0.1	99.5 / 0.3	97.8 / 0.1	99.9 / 0.0
walker walk	97.7 / 0.2	98.8 / 0.1	97.5 / 0.2	99.7 / 0.1	99.1 / 0.2	99.7 / 0.1	100.0 / 0.1	100.1 / 0.1	98.5 / 0.1	100.4 / 0.1
total	373.6	363.2	353.3	363.3	367.3	367.7	207.6	215.0	373.1	368.9

Table 14. Mixed Objective Dataset: best scores with large architecture and 5 critics

					mean/stdErr					
agent	IQL		ASAC		AWAC		TD3		TD3+BC	
actor dim	1024		1024		1024		1024		1024	
value archi dim	1024		0		0		0		0	
num cri	5		5		5		5		5	
critic num layers	2		2		2		2		2	
actor num layers	2		2		2		2		2	
critic model	modern		modern		modern		modern		modern	
actor model	modern		modern		modern		modern		modern	
value archi model	simple		not avail		not avail		not avail		not avail	
score processing	softmax		asac buffer		softmax		none		none	
load replay buffer	mixed obj. data		mixed obj. data		mixed obj. data		mixed obj. data		mixed obj. data	
num es	1	50	1	50	1	50	1	50	1	50
task										
cheetah	93.2	89.0	94.7	94.3	95.0	93.8	82.2	82.2	93.3	91.5
cheetah run	83.0 / 1.0	62.1 / 3.5	86.2 / 1.1	82.0 / 0.7	86.4 / 0.2	80.1 / 2.1	34.7 / 1.2	35.6 / 2.9	75.7 / 0.8	69.0 / 2.9
cheetah run backward	90.3 / 0.6	94.4 / 0.2	93.5 / 0.3	95.6 / 0.1	94.0 / 0.3	95.4 / 0.2	94.5 / 0.4	93.7 / 0.5	97.7 / 0.1	97.1 / 0.5
cheetah walk	99.9 / 0.0	99.8 / 0.1	99.7 / 0.1	99.8 / 0.0	99.9 / 0.0	99.8 / 0.0	99.9 / 0.0	100.0 / 0.0	100.0 / 0.0	100.0 / 0.0
cheetah walk backward	99.7 / 0.0	99.7 / 0.0	99.4 / 0.2	99.7 / 0.0	99.6 / 0.0	99.7 / 0.0	99.7 / 0.0	99.7 / 0.0	99.7 / 0.0	99.7 / 0.0
humanoid	90.9	78.2	90.4	79.1	91.4	78.5	3.1	2.5	83.4	76.4
humanoid run	97.0 / 0.7	64.9 / 0.7	89.6 / 1.0	62.9 / 0.5	90.5 / 0.2	59.5 / 0.7	2.5 / 0.4	2.6 / 0.4	79.1 / 1.2	56.3 / 0.5
humanoid stand	82.4 / 0.5	79.2 / 0.2	87.8 / 0.4	83.1 / 0.6	89.2 / 0.5	84.8 / 0.4	4.4 / 0.6	3.1 / 1.0	80.9 / 1.3	82.3 / 0.2
humanoid walk	93.2 / 0.3	90.4 / 0.3	93.9 / 0.1	91.2 / 0.1	94.3 / 0.1	91.1 / 0.3	2.3 / 0.5	1.7 / 0.3	90.3 / 0.4	90.5 / 0.2
quadruped	91.8	96.7	83.2	92.2	85.4	92.2	53.4	57.9	91.5	93.8
quadruped jump	89.1 / 1.0	96.0 / 0.1	79.6 / 3.7	94.1 / 0.4	81.0 / 3.3	94.5 / 1.0	62.6 / 3.1	68.5 / 7.1	90.4 / 1.3	92.9 / 0.8
quadruped run	94.8 / 0.6	98.9 / 0.5	81.0 / 3.6	86.7 / 1.9	95.2 / 1.6	87.2 / 5.4	48.7 / 1.1	49.2 / 1.5	92.7 / 1.2	97.2 / 1.2
quadruped stand	94.2 / 0.8	95.7 / 0.3	83.6 / 1.6	95.2 / 0.7	81.6 / 0.5	94.7 / 0.9	63.8 / 2.8	66.8 / 4.2	88.3 / 3.2	91.3 / 2.5
quadruped walk	89.2 / 1.2	96.1 / 0.3	88.8 / 1.0	92.7 / 0.8	83.8 / 5.6	92.6 / 1.9	38.4 / 1.6	47.2 / 2.3	94.5 / 0.9	93.9 / 1.8
walker	94.6	95.3	96.5	96.7	95.8	95.7	94.0	94.8	98.2	98.1
walker run	84.1 / 0.4	86.1 / 1.7	93.9 / 0.4	89.7 / 0.6	88.1 / 1.5	85.2 / 1.3	76.9 / 2.4	80.1 / 3.0	94.5 / 0.3	94.2 / 0.5
walker spin	98.3 / 0.1	98.7 / 0.0	94.8 / 0.4	98.5 / 0.1	97.5 / 0.2	98.7 / 0.1	99.3 / 0.0	99.3 / 0.0	99.1 / 0.1	98.9 / 0.1
walker stand	97.8 / 0.2	98.4 / 0.0	98.7 / 0.0	99.3 / 0.0	98.7 / 0.1	99.3 / 0.1	99.7 / 0.0	99.7 / 0.0	99.3 / 0.0	99.4 / 0.1
walker walk	98.4 / 0.1	98.0 / 0.2	98.7 / 0.2	99.4 / 0.1	98.9 / 0.2	99.5 / 0.1	100.2 / 0.2	100.2 / 0.0	100.0 / 0.0	99.7 / 0.2
total	370.6	359.2	364.9	362.3	367.5	360.2	232.7	237.5	366.4	359.7

Table 15. Mixed Objective Dataset: best scores with modern architecture and 5 critics

C.2. Locomotion-v2: Full Results

	mean/stdErr			
agent	IQL	AWAC	TD3	TD3+BC
actor dim	256	256	256	256
num cri	2	2	2	2
critic num layers	2	2	2	2
actor num layers	2	2	2	2
critic model	simple	simple	simple	simple
actor model	simple	simple	simple	simple
num es	1	1	1	1
task				
halfcheetah-medium-expert-v2	87.6 / 1.2	93.0 / 0.5	3.9 / nan	91.3 / 0.4
halfcheetah-medium-replay-v2	45.0 / 0.2	48.0 / 0.2	42.0 / 2.8	47.6 / 0.2
halfcheetah-medium-v2	51.6 / 0.1	51.9 / 0.2	44.9 / 7.4	66.2 / 0.2
hopper-medium-expert-v2	107.9 / 0.6	100.8 / 1.8	0.7 / 0.0	95.5 / 4.0
hopper-medium-replay-v2	96.7 / 0.7	99.1 / 1.0	64.8 / 21.3	79.8 / 4.3
hopper-medium-v2	97.4 / 3.2	79.6 / 1.6	NaN	72.9 / 18.1
walker2d-medium-expert-v2	112.2 / 0.9	109.7 / 0.1	-0.1 / 0.0	108.9 / 2.7
walker2d-medium-replay-v2	88.0 / 1.5	82.6 / 1.7	3.5 / 0.6	75.7 / 5.1
walker2d-medium-v2	85.4 / 1.3	83.5 / 2.4	-0.1 / 0.1	80.7 / 0.1
total	771.8	748.2	159.7	718.6

Table 16. Locomotion-v2: best scores with small architecture

	mean/stdErr									
agent	IQL		ASAC		AWAC		TD3		TD3+BC	
actor dim	1024		1024		1024		1024		1024	
num cri	5		5		5		5		5	
critic num layers	3		3		3		3		3	
actor num layers	5		5		5		5		5	
critic model	simple		simple		simple		simple		simple	
actor model	simple		simple		simple		simple		simple	
num es	1	50	1	50	1	50	1	50	1	50
task										
halfcheetah-medium-expert-v2	93.5 / 0.2	60.7 / 1.4	74.1 / 18.7	88.5 / 1.2	90.5 / 0.7	100.3 / 1.0	0.3 / 2.9	15.5 / 13.1	91.8 / 0.4	94.1 / 0.8
halfcheetah-medium-replay-v2	47.4 / 0.2	52.0 / 0.2	47.9 / 0.3	54.9 / 0.3	50.4 / 0.2	57.5 / 0.1	45.1 / 2.2	54.5 / 2.2	49.8 / 0.8	59.1 / 0.6
halfcheetah-medium-v2	54.9 / 0.1	63.0 / 0.1	49.4 / 0.2	61.1 / 0.2	53.4 / 0.2	64.6 / 0.2	27.9 / 17.7	20.3 / 11.5	55.8 / 0.1	70.8 / 0.4
hopper-medium-expert-v2	110.0 / 0.2	30.2 / 2.6	109.8 / 0.2	64.5 / 8.3	110.4 / 0.2	65.8 / 8.9	0.9 / 0.2	1.0 / 0.2	109.6 / 0.4	71.3 / 8.8
hopper-medium-replay-v2	94.7 / 1.9	85.7 / 4.5	72.9 / 4.9	95.2 / 1.4	90.8 / 2.7	102.5 / 0.5	12.9 / 5.4	29.6 / 13.0	60.0 / 4.3	98.7 / 2.1
hopper-medium-v2	96.4 / 2.0	67.4 / 13.9	75.6 / 1.6	57.3 / 3.6	81.7 / 16.2	55.3 / 7.4	0.9 / 0.2	0.8 / 0.0	93.2 / 0.7	59.9 / 2.9
walker2d-medium-expert-v2	110.3 / 0.1	114.1 / 0.4	110.7 / 0.0	113.5 / 0.3	110.1 / 0.1	115.5 / 0.3	0.2 / 0.4	1.0 / 0.6	110.8 / 0.1	115.5 / 0.9
walker2d-medium-replay-v2	88.7 / 1.4	94.2 / 0.7	65.5 / 2.8	91.1 / 1.7	85.3 / 0.7	96.3 / 1.0	3.8 / 0.6	2.9 / 0.9	67.1 / 1.1	87.7 / 0.4
walker2d-medium-v2	84.3 / 2.2	89.8 / 0.4	77.2 / 0.4	84.6 / 0.2	82.6 / 0.1	87.0 / 0.3	0.1 / 0.3	0.9 / 0.8	80.6 / 0.3	87.2 / 0.3
total	780.1	657.2	683.2	710.7	755.2	744.7	92.2	126.5	718.8	744.4

Table 17. Locomotion-v2: best scores with large architecture

	mean/stdErr									
agent	IQL		ASAC		AWAC		TD3		TD3+BC	
actor dim	1024		1024		1024		1024		1024	
num cri	5		5		5		5		5	
critic num layers	2		2		2		2		2	
actor num layers	2		2		2		2		2	
critic model	modern		modern		modern		modern		modern	
actor model	modern		modern		modern		modern		modern	
num es	1	50	1	50	1	50	1	50	1	50
task										
halfcheetah-medium-expert-v2	93.4 / 0.1	44.2 / 1.0	94.5 / 0.2	76.6 / 2.8	94.3 / 0.4	63.0 / 1.0	29.3 / 3.4	27.7 / 4.6	92.7 / 0.2	79.2 / 0.9
halfcheetah-medium-replay-v2	46.5 / 0.2	49.9 / 0.1	46.7 / 0.6	54.5 / 0.2	49.0 / 0.2	54.4 / 0.3	52.1 / 0.6	52.6 / 1.1	55.9 / 0.5	57.9 / 0.1
halfcheetah-medium-v2	52.9 / 0.2	58.2 / 0.1	52.9 / 0.2	61.3 / 0.5	53.1 / 0.2	61.4 / 0.1	62.2 / 1.1	63.0 / 0.3	63.4 / 0.2	64.6 / 0.3
hopper-medium-expert-v2	110.5 / 0.2	29.1 / 4.6	110.9 / 0.2	45.5 / 13.4	110.3 / 0.1	62.7 / 13.5	1.6 / 0.3	2.8 / 1.4	111.8 / 0.3	81.2 / 2.6
hopper-medium-replay-v2	100.0 / 0.8	101.1 / 0.5	84.7 / 4.6	101.4 / 0.4	101.0 / 0.4	101.9 / 0.6	35.2 / 1.3	40.6 / 12.8	72.1 / 2.9	100.6 / 0.2
hopper-medium-v2	98.1 / 0.8	73.0 / 6.1	89.1 / 2.8	94.6 / 4.7	95.1 / 2.3	96.4 / 2.8	1.3 / 0.3	1.7 / 0.2	94.9 / 0.8	88.3 / 2.0
walker2d-medium-expert-v2	111.4 / 0.6	112.9 / 0.3	109.8 / 0.0	112.1 / 0.2	111.9 / 1.0	112.8 / 0.1	-0.2 / 0.0	-0.2 / 0.0	110.9 / 0.0	115.0 / 0.4
walker2d-medium-replay-v2	91.2 / 0.3	96.2 / 0.2	87.1 / 0.3	93.9 / 0.5	90.8 / 1.3	94.7 / 0.4	5.4 / 2.4	6.3 / 1.7	68.4 / 0.5	90.5 / 0.3
walker2d-medium-v2	89.3 / 0.6	90.4 / 0.2	81.1 / 0.2	86.1 / 0.1	83.5 / 0.2	87.1 / 0.1	-0.2 / 0.0	-0.2 / 0.0	79.1 / 0.5	87.1 / 0.2
total	793.2	655.1	756.8	726.0	789.0	734.2	186.7	194.3	749.1	764.3

Table 18. Locomotion-v2: best scores with modern architecture

C.3. Antmaze-v0: Full Results

	mean/stdErr		
agent	IQL	AWAC	TD3+BC
actor dim	256	256	256
num cri	2	2	2
critic num layers	2	2	2
actor num layers	2	2	2
critic model	simple	simple	simple
actor model	simple	simple	simple
score processing	softmax	softmax	none
num es	1	1	1
task			
antmaze-large-diverse-v0	41.5 / 3.4	36.5 / 2.2	2.1 / 1.7
antmaze-large-play-v0	28.7 / 9.0	7.8 / 7.8	0.2 / 0.2
antmaze-medium-diverse-v0	60.7 / 7.9	31.4 / 19.3	43.9 / 3.0
antmaze-medium-play-v0	66.3 / 2.1	76.3 / 1.6	28.1 / 11.7
antmaze-umaze-diverse-v0	68.8 / 7.3	14.0 / 14.0	41.4 / 3.3
antmaze-umaze-v0	89.1 / 3.4	45.6 / 15.0	70.2 / 15.8
total	355.2	211.6	186.0

Table 19. Antmaze-v0: best scores with small architecture

	mean/stdErr							
agent	IQL		ASAC		AWAC		TD3+BC	
actor dim	1024		1024		1024		1024	
num cri	5		5		5		5	
critic num layers	3		3		3		3	
actor num layers	5		5		5		5	
critic model	simple		simple		simple		simple	
actor model	simple		simple		simple		simple	
score processing	softmax		asac buffer		softmax		none	
num es	1	50	1	50	1	50	1	50
task								
antmaze-large-diverse-v0	59.8 / 3.2	64.4 / 1.3	10.3 / 1.4	49.2 / 0.8	16.9 / 10.6	32.0 / 1.1	7.2 / 3.9	8.7 / 1.3
antmaze-large-play-v0	26.2 / 2.3	22.6 / 3.7	4.7 / 2.2	12.3 / 1.0	7.2 / 4.6	17.3 / 0.7	0.6 / 0.6	5.6 / 3.5
antmaze-medium-diverse-v0	77.0 / 2.0	85.6 / 0.6	63.8 / 1.6	89.1 / 1.5	48.0 / 19.6	85.8 / 1.1	17.7 / 7.2	61.9 / 14.1
antmaze-medium-play-v0	78.5 / 1.9	86.5 / 0.7	51.5 / 4.4	86.7 / 1.2	63.3 / 15.9	84.6 / 1.0	15.7 / 11.1	24.2 / 10.0
antmaze-umaze-diverse-v0	82.3 / 3.6	83.9 / 0.9	71.1 / 2.2	72.1 / 1.5	48.5 / 10.3	42.0 / 17.8	69.4 / 3.7	68.1 / 5.4
antmaze-umaze-v0	91.0 / 0.7	96.6 / 0.4	91.2 / 2.1	99.0 / 0.2	89.8 / 1.4	98.2 / 0.3	89.7 / 4.1	96.0 / 1.2
total	414.8	439.6	292.6	408.5	273.7	359.8	200.3	264.6

Table 20. Antmaze-v0: best scores with 5 critics and large architecture

	mean/stdErr							
agent	IQL		ASAC		AWAC		TD3+BC	
actor dim	1024		1024		1024		1024	
num cri	5		5		5		5	
critic num layers	2		2		2		2	
actor num layers	2		2		2		2	
critic model	modern		modern		modern		modern	
actor model	modern		modern		modern		modern	
score processing	softmax		asac buffer		softmax		none	
num es	1	50	1	50	1	50	1	50
task								
antmaze-large-diverse-v0	42.1 / 1.7	60.4 / 1.2	40.0 / 1.6	49.9 / 1.5	43.0 / 2.2	43.0 / 3.4	18.8 / 1.5	9.9 / 9.9
antmaze-large-play-v0	38.9 / 1.7	54.2 / 1.1	33.4 / 2.4	43.8 / 2.4	32.0 / 1.8	42.2 / 0.8	7.2 / 4.3	8.3 / 5.4
antmaze-medium-diverse-v0	73.2 / 1.4	82.9 / 1.1	78.3 / 1.1	82.3 / 0.8	77.0 / 0.6	83.2 / 1.5	65.7 / 2.5	75.0 / 7.5
antmaze-medium-play-v0	71.0 / 1.5	82.9 / 0.7	74.1 / 1.3	84.9 / 1.4	77.0 / 1.4	85.6 / 1.7	70.5 / 1.7	52.4 / 17.6
antmaze-umaze-diverse-v0	77.3 / 3.5	86.6 / 0.7	65.7 / 3.3	81.1 / 2.9	62.5 / 2.6	72.8 / 3.3	67.7 / 7.8	71.7 / 4.9
antmaze-umaze-v0	88.6 / 1.0	96.1 / 0.7	95.2 / 0.8	98.7 / 0.1	96.9 / 0.7	97.8 / 0.5	74.6 / 18.6	94.1 / 1.2
total	391.1	463.1	386.7	440.8	388.3	424.8	304.3	311.5

Table 21. Antmaze-v0: best scores with 5 critics and modern architecture

	mean/stdErr							
agent	IQL		ASAC		AWAC		TD3+BC	
actor dim	1024		1024		1024		1024	
num cri	10		10		10		10	
critic num layers	3		3		3		3	
actor num layers	5		5		5		5	
critic model	simple		simple		simple		simple	
actor model	simple		simple		simple		simple	
score processing	softmax		asac buffer		softmax		none	
num es	1	50	1	50	1	50	1	50
task								
antmaze-large-diverse-v0	57.4 / 1.6	63.4 / 1.7	12.0 / 1.0	52.4 / 3.0	34.8 / 7.2	44.8 / 6.7	2.8 / 2.4	4.4 / 2.3
antmaze-large-play-v0	27.3 / 1.3	13.8 / 2.1	8.1 / 0.3	12.6 / 1.7	14.1 / 7.1	12.5 / 1.1	0.7 / 0.3	0.0 / 0.0
antmaze-medium-diverse-v0	78.3 / 1.3	80.2 / 1.9	65.6 / 1.6	88.7 / 0.6	83.7 / 1.4	87.1 / 0.9	12.4 / 0.6	17.3 / 1.3
antmaze-medium-play-v0	77.0 / 1.0	86.6 / 1.2	59.2 / 1.4	87.2 / 1.4	67.0 / 6.1	85.2 / 0.7	16.5 / 0.9	9.2 / 9.2
antmaze-umaze-diverse-v0	84.3 / 2.1	87.5 / 0.6	77.9 / 6.7	79.3 / 1.7	39.4 / 15.0	39.6 / 15.7	53.3 / 14.8	40.9 / 10.3
antmaze-umaze-v0	93.2 / 1.3	95.8 / 1.0	88.7 / 0.4	98.6 / 0.4	91.1 / 0.4	98.9 / 0.2	97.6 / 0.4	99.2 / 0.3
total	417.5	427.3	311.4	418.9	330.2	368.0	183.4	170.9

Table 22. Antmaze-v0: best scores with 10 critics and large architecture

	mean/stdErr							
agent	IQL		ASAC		AWAC		TD3+BC	
actor dim	1024		1024		1024		1024	
num cri	10		10		10		10	
critic num layers	2		2		2		2	
actor num layers	2		2		2		2	
critic model	modern		modern		modern		modern	
actor model	modern		modern		modern		modern	
score processing	softmax		asac buffer		softmax		none	
num es	1	50	1	50	1	50	1	50
task								
antmaze-large-diverse-v0	40.8 / 1.3	61.7 / 1.3	42.8 / 3.6	50.9 / 1.2	44.3 / 1.3	52.4 / 3.3	11.2 / 1.7	6.5 / 3.7
antmaze-large-play-v0	40.1 / 1.2	51.4 / 1.6	36.1 / 1.9	43.3 / 1.4	31.4 / 1.6	39.9 / 1.1	0.8 / 0.5	0.0 / 0.0
antmaze-medium-diverse-v0	74.9 / 1.5	83.1 / 0.7	77.1 / 2.0	83.3 / 1.3	79.2 / 0.7	84.7 / 0.6	38.4 / 15.7	27.3 / 16.7
antmaze-medium-play-v0	65.8 / 4.5	81.4 / 1.3	77.7 / 1.4	83.6 / 1.4	79.3 / 1.2	86.2 / 0.8	68.2 / 1.6	68.0 / 1.4
antmaze-umaze-diverse-v0	71.6 / 2.1	82.3 / 0.7	63.5 / 5.6	75.5 / 1.7	86.7 / 1.0	74.4 / 2.8	83.5 / 5.9	64.1 / 17.5
antmaze-umaze-v0	87.3 / 0.6	96.2 / 0.3	94.8 / 0.6	98.8 / 0.2	97.8 / 0.1	98.7 / 0.3	96.8 / 0.3	99.3 / 0.1
total	380.4	456.1	391.9	435.4	418.7	436.1	299.0	265.1

Table 23. Antmaze-v0: best scores with 10 critics and modern architecture

C.4. Locomotion-0.1-v2: Full Results

	mean/stdErr							
agent	IQL		ASAC		AWAC		TD3+BC	
actor dim	1024		1024		1024		1024	
num cri	10		10		10		10	
critic num layers	3		3		3		3	
actor num layers	5		5		5		5	
critic model	simple		simple		simple		simple	
actor model	simple		simple		simple		simple	
score processing	softmax		asac buffer		softmax		none	
num es	1	50	1	50	1	50	1	50
task								
halfcheetah-random-expert-0.1-v2	76.3 / 2.1	19.7 / 0.8	61.5 / 6.7	27.0 / 1.2	78.3 / 1.2	31.5 / 1.1	29.7 / 1.1	30.7 / 1.2
halfcheetah-random-medium-0.1-v2	48.8 / 0.2	52.7 / 0.2	45.2 / 0.2	53.6 / 4.1	51.9 / 0.2	58.9 / 0.2	59.4 / 0.6	60.7 / 0.7
hopper-random-expert-0.1-v2	95.4 / 2.2	24.3 / 2.2	41.8 / 2.7	20.4 / 1.0	78.4 / 1.5	26.9 / 1.3	77.0 / 2.0	26.5 / 1.8
hopper-random-medium-0.1-v2	70.5 / 3.0	64.1 / 4.6	48.0 / 5.0	66.7 / 2.3	68.7 / 4.1	76.3 / 3.8	46.6 / 1.8	60.5 / 1.7
walker2d-random-expert-0.1-v2	99.2 / 0.6	21.7 / 0.9	101.1 / 0.6	60.3 / 3.5	61.3 / 15.5	40.0 / 10.2	3.8 / 0.3	11.6 / 0.8
walker2d-random-medium-0.1-v2	73.3 / 0.6	60.5 / 1.6	71.4 / 1.6	75.0 / 0.9	59.3 / 14.7	78.4 / 3.1	53.1 / 10.4	48.6 / 19.6
total	463.5	242.9	369.0	303.1	398.0	311.8	269.5	238.6

Table 24. Locomotion-0.1-v2: best scores with 10 critics and large architecture

	mean/stdErr							
agent	IQL		ASAC		AWAC		TD3+BC	
actor dim	1024		1024		1024		1024	
num cri	10		10		10		10	
critic num layers	2		2		2		2	
actor num layers	2		2		2		2	
critic model	modern		modern		modern		modern	
actor model	modern		modern		modern		modern	
score processing	softmax		asac buffer		softmax		none	
num es	1	50	1	50	1	50	1	50
task								
halfcheetah-random-expert-0.1-v2	65.8 / 2.2	12.3 / 0.8	56.2 / 6.6	25.8 / 0.6	66.7 / 3.0	27.5 / 0.8	15.5 / 1.2	18.3 / 1.2
halfcheetah-random-medium-0.1-v2	47.3 / 0.4	48.9 / 0.3	47.4 / 0.5	53.2 / 0.4	51.0 / 0.2	54.6 / 0.1	56.5 / 0.5	57.5 / 0.3
hopper-random-expert-0.1-v2	109.0 / 0.4	40.2 / 1.6	73.4 / 5.3	29.6 / 2.2	32.3 / 4.8	25.8 / 2.0	80.2 / 4.4	32.1 / 1.9
hopper-random-medium-0.1-v2	93.9 / 2.3	92.3 / 1.6	52.7 / 1.2	69.0 / 2.1	53.4 / 1.0	71.7 / 1.2	50.8 / 1.0	71.1 / 1.9
walker2d-random-expert-0.1-v2	107.8 / 0.3	44.4 / 3.6	99.5 / 2.0	61.4 / 4.1	93.6 / 3.1	83.9 / 4.2	41.9 / 17.0	34.3 / 11.8
walker2d-random-medium-0.1-v2	75.0 / 0.5	65.6 / 2.1	55.4 / 5.1	65.9 / 4.7	30.9 / 13.0	47.0 / 6.9	52.6 / 4.0	57.5 / 4.3
total	498.8	303.7	384.5	304.9	327.9	310.4	297.5	270.8

Table 25. Locomotion-0.1-v2: best scores with 10 critics and modern architecture

C.5. ExoRL: Full Results

We also tested our hypotheses on the ExoRL dataset generated using RND (Yarats et al., 2022).

	mean/stdErr			
agent	IQL	AWAC	TD3	TD3+BC
actor dim	256	256	256	256
num cri	2	2	2	2
critic num layers	2	2	2	2
actor num layers	2	2	2	2
critic model	simple	simple	simple	simple
actor model	simple	simple	simple	simple
num es	1	1	1	1
task				
cheetah	47.1	54.1	54.1	64.7
cheetah run	16.4 / 0.3	19.0 / 0.6	23.8 / 6.2	29.5 / 2.5
cheetah run backward	29.2 / 2.9	40.5 / 4.6	56.0 / 1.7	57.9 / 3.6
cheetah walk	56.0 / 2.3	66.6 / 3.2	68.9 / 7.6	73.2 / 3.2
cheetah walk backward	86.9 / 2.8	90.4 / 6.2	67.8 / 31.0	98.1 / 0.8
quadruped	82.1	84.8	84.2	86.6
quadruped jump	88.5 / 0.8	89.0 / 0.6	95.4 / 1.5	93.1 / 0.6
quadruped run	66.3 / 0.3	64.7 / 0.8	68.2 / 1.5	68.9 / 0.5
quadruped stand	99.2 / 0.2	98.5 / 0.5	97.7 / 0.4	99.5 / 0.2
quadruped walk	74.6 / 0.3	86.9 / 2.9	75.6 / 4.6	84.7 / 2.3
walker	55.0	50.9	79.0	73.8
walker run	17.0 / 0.1	17.2 / 0.1	46.2 / 1.0	41.4 / 0.4
walker spin	92.5 / 0.3	93.5 / 0.5	99.2 / 0.1	98.7 / 0.1
walker stand	65.9 / 1.3	48.2 / 0.5	92.0 / 2.3	83.4 / 1.1
walker walk	44.7 / 0.2	44.6 / 1.4	78.7 / 1.6	71.7 / 1.8
total	184.3	189.8	217.4	225.0

Table 26. Exorl: best scores with small network

	mean/stdErr							
agent	IQL		AWAC		TD3		TD3+BC	
actor dim	1024		1024		1024		1024	
num cri	10		10		10		10	
critic num layers	3		3		3		3	
actor num layers	5		5		5		5	
critic model	simple		simple		simple		simple	
actor model	simple		simple		simple		simple	
num es	1	50	1	50	1	50	1	50
task								
cheetah	54.8	61.6	63.2	71.0	70.5	74.1	80.0	80.2
cheetah run	24.5 / 0.4	29.0 / 0.3	22.5 / 0.2	27.4 / 0.4	51.5 / 0.4	52.4 / 0.7	54.0 / 0.5	53.7 / 0.3
cheetah run backward	28.6 / 0.6	42.6 / 1.0	42.0 / 0.7	60.1 / 0.9	67.6 / 1.9	68.3 / 0.9	71.6 / 1.5	70.2 / 1.3
cheetah walk	70.5 / 0.3	77.2 / 0.7	90.2 / 0.6	97.2 / 0.2	94.2 / 0.6	95.5 / 0.7	94.7 / 0.5	97.1 / 0.6
cheetah walk backward	95.5 / 0.6	97.7 / 0.3	98.4 / 0.2	99.4 / 0.1	68.7 / 17.7	80.2 / 11.0	99.5 / 0.0	99.6 / 0.0
quadruped	80.7	81.9	84.4	83.3	46.7	44.8	81.7	81.8
quadruped jump	93.9 / 0.3	96.4 / 0.3	93.0 / 0.4	94.9 / 0.8	54.3 / 15.0	51.0 / 11.0	97.1 / 0.1	99.2 / 0.1
quadruped run	61.6 / 0.1	62.1 / 0.2	61.6 / 0.4	61.2 / 0.2	44.2 / 7.4	41.1 / 8.3	62.4 / 0.1	63.6 / 0.3
quadruped stand	100.6 / 0.1	100.8 / 0.0	100.0 / 0.2	100.5 / 0.1	65.6 / 10.5	63.7 / 11.1	100.0 / 0.1	100.2 / 0.1
quadruped walk	66.8 / 0.9	68.2 / 2.1	83.2 / 1.9	76.7 / 2.0	22.8 / 6.0	23.6 / 4.0	67.2 / 1.8	64.0 / 2.0
walker	64.6	71.2	67.2	74.2	63.3	66.8	84.0	81.5
walker run	19.2 / 0.1	27.7 / 0.0	20.4 / 0.0	31.7 / 0.0	37.2 / 2.9	39.8 / 2.1	53.0 / 0.6	47.7 / 0.6
walker spin	97.6 / 0.1	98.6 / 0.0	98.4 / 0.1	99.4 / 0.0	95.2 / 1.1	97.8 / 0.7	99.0 / 0.0	99.3 / 0.0
walker stand	79.6 / 0.2	86.7 / 0.2	79.6 / 0.6	88.1 / 0.5	62.3 / 5.4	68.3 / 10.4	94.0 / 0.1	94.9 / 0.1
walker walk	61.9 / 0.1	71.6 / 0.3	70.3 / 0.3	77.5 / 0.4	58.5 / 2.5	61.3 / 6.1	90.2 / 0.8	83.9 / 1.3
total	200.0	214.6	214.8	228.5	180.5	185.7	245.7	243.4

Table 27. Exorl: best scores with 10 critics and large simple architecture

C.6. Antmaze-v2 results with IQL

C.7. D4RL locomotion results with ReBRAC: layer normalization and large batch size

We examine the increased batch size and layer norm tricks proposed in concurrent work (ReBRAC, (Tarasov et al., 2024)), adding to our AWAC and IQL implementations with the proposed scaling strategy. We note that both these practices considerably increase both computational demand and parameter count, yet do not seem to provide consistent and significant benefits. (Table 29).

	mean/stdErr	
agent	IQL	
actor dim	1024	
num cri	5	
critic model	modern	
actor model	modern	
num es	1	50
antmaze-large-diverse-v2	47.6 / 3.1	65.2 / 3.6
antmaze-large-play-v2	43.2 / 5.5	59.2 / 3.4
antmaze-medium-diverse-v2	73.6 / 3.7	82.0 / 1.9
antmaze-medium-play-v2	66.8 / 1.0	84.4 / 3.1
antmaze-umaze-diverse-v2	75.6 / 5.6	89.2 / 1.4
antmaze-umaze-v2	93.2 / 2.8	96.8 / 0.5
total	400.0	476.8

Table 28. Antmaze-v2: best scores with modern models and 5 critics, with and without ES. These preliminary results appear to match the Antmaze-v0 results, indicating the changes between these versions are not particularly significant.

	mean/stdErr			
agent	IQL		AWAC	
actor dim	1024		1024	
num cri	5		5	
critic num layers	3		3	
actor num layers	5		5	
critic model	simple		simple	
actor model	simple		simple	
num es	1	50	1	50
task				
halfcheetah-medium-expert-v2	93.9 / 0.2	67.9 / 2.2	94.3 / 0.4	107.2 / 0.3
halfcheetah-medium-replay-v2	44.5 / 0.1	46.5 / 0.4	48.0 / 0.2	52.1 / 0.3
halfcheetah-medium-v2	49.7 / 0.1	57.3 / 0.2	49.6 / 0.1	61.3 / 0.3
hopper-medium-expert-v2	110.7 / 0.4	29.1 / 3.2	110.9 / 0.2	102.7 / 4.1
hopper-medium-replay-v2	100.1 / 0.1	96.7 / 2.6	94.7 / 2.6	103.6 / 0.2
hopper-medium-v2	76.8 / 1.9	88.0 / 1.0	81.0 / 2.6	98.0 / 1.2
walker2d-medium-expert-v2	109.5 / 0.1	111.8 / 0.3	109.8 / 0.0	112.9 / 0.7
walker2d-medium-replay-v2	87.1 / 2.3	96.0 / 0.3	85.1 / 1.4	95.7 / 0.6
walker2d-medium-v2	84.5 / 0.5	89.1 / 0.5	87.1 / 2.3	94.5 / 0.9
total	756.9	682.4	760.7	827.9

Table 29. D4RL locmotion: best scores with large models, 5 critics, layer normalization and batch-size=1024.

Algorithm 3 Logsumexp-tree updating

input leaf node n
input unnormalized log probability l
n.value $\leftarrow l$
while not $n =$ root **do**
  $n \leftarrow n$.parent
  $a \leftarrow n$.leftchild.value
  $b \leftarrow n$.rightchild.value
  $m \leftarrow \max(a, b)$
  n.value $\leftarrow m + \text{log1p}(e^{a-m} + e^{b-m})$
end while

Algorithm 4 Logsumexp-tree sampling

Sample $u \sim U[0, 1]$, $r \leftarrow \log u +$ root.value, $n \leftarrow$ root
while not leaf(n) **do**
  $v \leftarrow n$.leftchild.value
  if if $r < v$ **then**
    $n \leftarrow n$.leftchild
  else
    $n \leftarrow n$.rightchild, $r \leftarrow r + \text{log1p}(-e^{v-r})$
  end if
end while
return n

D. ASAC implementation - logsumexp tree data-structure

Implementing our new ASAC algorithm requires sampling according to the action-maximizing constrained distribution given by $\mathcal{B}^*(s, a) := \frac{1}{Z}\mathcal{B}(s, a)\exp(A_\phi(s, a)/\beta)$ in an efficient and stable manner. This introduced several challenges, which our new *logsumexp-tree* data structure was designed to address.

Sum-tree summary. Sampling with unnormalized probabilities can be achieve with a traditional sum-tree. A sum-tree is a data-structure taking the form of a binary tree where the value of each of its nodes corresponds to the value of its children. Hence, we can store unnormalized probabilities in each of N leaf nodes, making the root correspond to the normalizing factor Z. Hence, every time an unnormalized probability is updated, we only require $O(\log N)$ iterative updates to recompute the values its ancestors, leaving the other nodes unmodified. Similarly, we can sample from the true distribution via sampling a uniform $r \sim U[0, Z]$ and do $O(\log N)$ comparisons until we reach one of the leaves. In particular, starting from the root, we compare r with a node's left child value v: if $v < r$ we descend to the left subtree, otherwise we descend to the right subtree and update $r \leftarrow r - v$. Schaul et al. (2015) slightly modify the first step of this procedure when sampling an n-sized minibatch, by dividing Z into n equal-length segments and obtaining each $r_i \in \{r_{1:n}\}$ by sampling from $U[Z/n \times (i-1), Z/n \times i]$. This is done to collect more 'spread-out' samples across each minibatch, something that we found did not seem to play a significant effect on bias or performance.

Logsumexp-tree. In our use case, the unnormalized probablities given by $\mathcal{B}$ (Equation 6) are the result of a scaled exponentiation whose magnitude appears to notably vary across problem setting and training stage. Hence, in practice, we found that directly recording $\mathcal{B}^*(s, a)$ into a sum-tree resulted in arithmetic underflow (with many of the leaves and their sums collapsing to zeros) and overflow (leading to crashes due to exceeding the maximum representable values). To address these challenges, the logsumexp-tree allows to record the unnormalized logits before exponentiation $q(s, a) = A_\phi(s, a)/\beta$. Moreover, it allows to perform updating and sampling operations with the same $O(\log n)$ complexity as a sum-tree with stable operations without having to store any explicit values outside log space. In particular, each node p in our new data stracture stores the 'logsumexp' of its children a, b, an operation that can be stably done via first shifting a and b by their maximum and using the highly-precise log1p operation implemented in Numpy/Pytorch. In a similar fashion, we can now sample by transforming a uniform variable and applying a log transformation. Hence, analogously to the sum-tree, starting with the root node, we can descend the logsumexp-tree by comparing our sample $r \in (-\infty, \log(Z)]$ with its left child's value v to choose which branch to follow. However, this time, if $r \geq v$ we perform a 'logsubstractexp' operation to update the value of r. We refer to Algorithms 3 and 4 for further details and the exact mathematical operations involved. In our shared code, we provide an implementation of the logsumexp-tree stored in a simple array representation, allowing for efficient fully-parallelized operations.

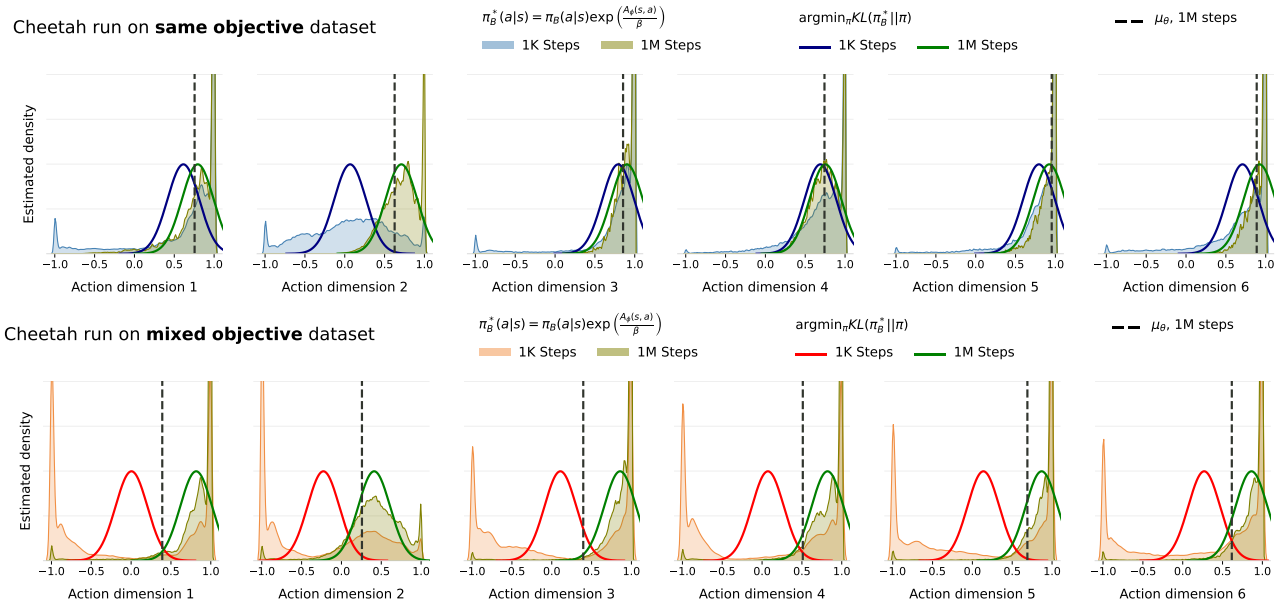

Figure 9. Estimated advantage-weighted action-distribution from $\pi_{\mathcal{B}}^*$ and the corresponding optimal projection with a Gaussian π after 1K and 1M optimization steps of offline training with AWAC on the cheetah run task with the same objective dataset (A) and the mixed objective dataset (B).

E. The Role of Mean-Seeking

One of our early hypotheses to what can cause over-conservativism was that Gaussian policies are a poor fit to multi-modal data distributions (e.g., as Fig. 3 clearly shows). Unfortunately, all our attempts to fix this issue, including the introduction of more expressive policy models (mixture of Gaussians, normalizing flows) or the usage of a reverse (mode-seeking) KL in the loss, led to performance collapse (Fig. 10). This failure was mostly due to early overfitting to the distribution matching objective while disregarding Q-function maximization, a phenomenon conceptually similar to posterior collapse in density modeling (Bowman et al., 2015; Kingma et al., 2016). This led us to discard such an hypothesis, as Gaussian policies seems to have a useful regularizing effect (at least early in training), and to focus only on ES as a solution to over-conservativism.

We focus on the role of mean-seeking and early attempts to overcome the action-averaging phenomenon illustrated in Figure 3. An extended version of such a figure with all action dimensions is shown in Figure 9. As detailed in the main text, the policy improvement optimization in TD3+BC, AWAC, and IQL can be be seen as performing a forward KL projection with respect to some inferred target distribution. Furthermore, they all parameterize a strictly unimodal policy, taking the form of either a Gaussian or a squashed Gaussian distribution. Hence, in case the target distribution displays significant multi-modality, using such models in conjunction with the mean-seeking nature of the forward KL loss would prevent ever closely matching part of the behavior data, leading to the displayed action-averaging phenomenon. Based on this considerations, we empirically analyze the effects of this induced mean-seeking regularization and the consequences from its relaxation.

We find that, in spite of such problematic side-effects, mean-seeking regularization appears to be playing a crucial role to avoid premature convergence to an early suboptimal equilibrium. In particular, we analyze the effects of replacing the Gaussian policy and the forward KL objective with alternatives that allow to relax or overcome the mean-seeking regime. First, we analyze two simple variations of TD3 by adding to its original policy improvement objective from Equation 2 an auxiliary term to maximize either $\mathbb{E}_{a\sim\pi'_{\mathcal{B}}(\cdot|s)}\left[\log \pi_\theta(a|s)\right]$ or $\mathbb{E}_{a'\sim\pi_\theta(\cdot|s)}\left[\log \pi'_{\mathcal{B}}(a'|s)\right]$ using the pre-trained behavior model $\pi'_{\mathcal{B}}$ described in Appendix E.1. We note that the first case is practically equivalent to the TD3+BC algorithm but with actions sampled from the learned behavior policy rather than the dataset, still falling in the mean-seeking regime. On the other hand, the auxiliary term in the second case corresponds to minimizing an *inverse* KL with the behavior policy, $D_{KL}(\pi_\theta|\pi'_{\mathcal{B}})$, and is akin to directly optimizing for the dual objective leading to AWAC's constrained advantage-maximizing targets $\pi_{\mathcal{B}}^*$ (Peters & Schaal, 2007). In principle, this latter approach should fully preserve the canonical support constraint at the basis of most offline RL algorithms without the detrimental action-averaging, as it would allow the agent to focus on a single subset of $\pi_{\mathcal{B}}^*$

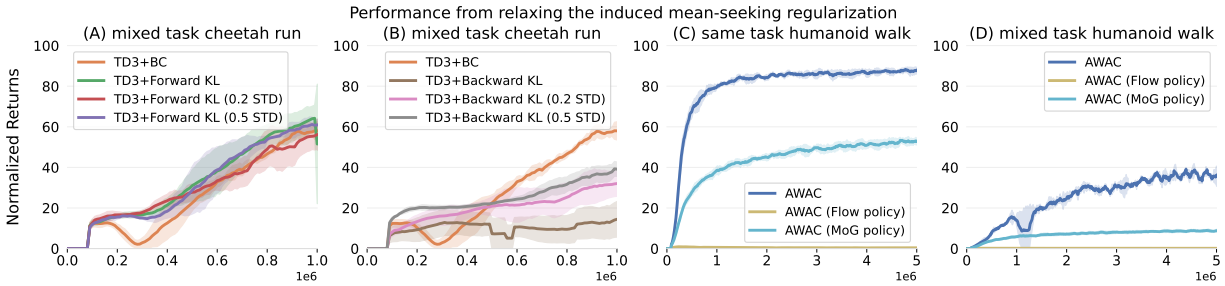

Figure 10. Results from using pre-trained density models (as described in Appendix E.1) to optimize for auxiliary policy improvement losses based on forward (A) and backward (B) KL divergences, and from increasing the policy's expressivity with affine Flows and Gaussian mixture distributions (C and D).

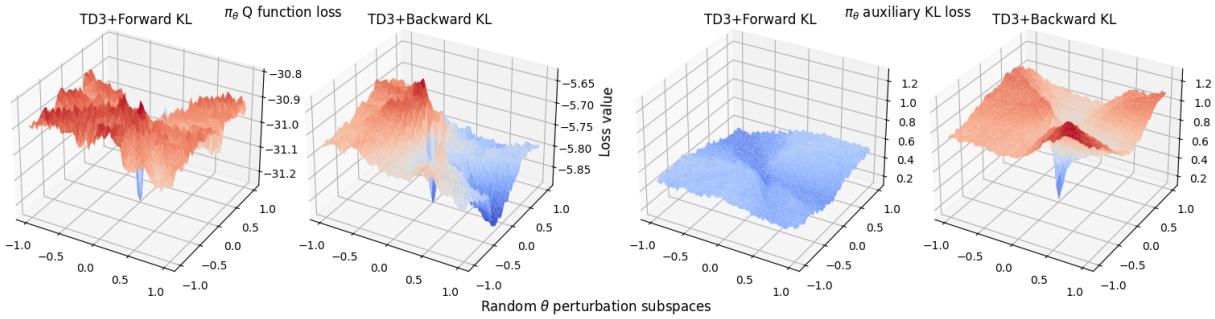

Figure 11. Two-dimensional loss surface projections using the visualization method from Li et al. (2018). We produce visualizations for the Q function maximization and auxiliary KL loss for both TD3 modifications with auxiliary forward and backward KL losses

right from the start.

As shown in part A of Figure 10, training with the mean-seeking forward KL objective expectedly yields very similar results to TD3+BC, validating the soundness of our formulated optimizations. However, when switching to the inverse KL objective in part B, we observe a severe degradation in performance, in direct contrast to its supposed theoretical benefits. We further analyze this phenomenon by pre-training increasingly mean-seeking proxies for $\pi_\mathcal{B}$ by fixing (rather than learning) the standard deviation of the output mixture distributions of $\pi'_\mathcal{B}$ to high values. The scope of these alternative parameterizations is to artificially emulate the mean-seeking regularization from behavior cloning and AWAC with the inverse KL objective, by forcing the pre-trained models themselves onto a mode-covering regime at the expense of accuracy. In particular, while the original $\pi'_\mathcal{B}$ with a learned standard deviation achieves a log-likelihood of 1.09 bits per dimension, fixing the standard deviations to 0.2 and 0.5 only attain log likelihoods of 0.01 and -0.54 bits per dimension, respectively. However, as shown in parts A and B, using these worse models paradoxically leads to higher performance, suggesting that the induced mean-seeking regularization is actually *an integral component* of current algorithms whose benefits far outweigh the downsides from modeling errors and action-averaging. As a validation check for our hypothesis, we also show that the performance of the forward KL TD3 variant, which already inherently encourages mean-seeking behavior regardless of the pre-trained models, does not improve and even slightly suffers from the resulting loss in precision. Finally, in parts C and D, we also analyze relaxing the mean-seeking regularization by increasing the expressivity of the policy's output distribution by parameterizing either a normalizing flow or a mixture of Gaussians. In particular, we consider either adding two additional affine flow layers conditioning on half the action dimensions and the state as in (Dinh et al., 2016) or enlarging the output of the policy to represent parameters for five Gaussian heads. While these policies even outperform their traditional unimodal counterpart in the online setting, when used by offline algorithms such as AWAC, they seem to produce visibly slower learning with an analogous collapse in final performance to the one observed with the inverse KL objective, occurring even when training for the less diverse same-objective datasets.

This observed performance stagnation suggests that optimizing the offline policy with a tractable unregularized distribution matching objective leads to a phenomenon analogous to posterior collapse in density modeling (Bowman et al., 2015;

Table 30. Autoregressive density model hyper-parameters used to obtain a proxy for the unknown behavior policy.

Density model hyper-parameters	
batch size	256
optimizer	Adam
learning rate	0.001
reserved validation data	15%
maximum epochs	100
encoder hidden layers	2
encoder hidden dimensionality	512
decoder hidden layers	2
decoder hidden dimensionality	$64 \times \lvert A \rvert$
decoder mixture components	10
non-linearity	ReLU

Kingma et al., 2016). In particular, relaxing canonical constrains appears to enable the agent to initially focus on matching a likely suboptimal subset of the behavior policy, incurring the risk of converging to an early equilibrium. Instead, the mean-seeking objective induced by combining unimodal policies with the forward KL minimization avoids this early collapsing pull but seems to incur in the aforementioned unwarranted mode-averaging. This phenomenon can also be qualitatively identified following Li et al. (2018), by visualizing the policy improvement loss surfaces induced by training with the TD3 variants employing the forward and backward KL auxiliary terms. As shown in Figure 11, the Q-function term of the actor loss attains a significantly lower absolute value and appears further from local convergence after training with the backward KL variant, reflecting its worse final performance. At the same time, the loss surface of the backward KL auxiliary term appears bound to a steep basin, in direct contrast with the much smoother minimum attained with its forward KL counterpart. Taken together, these visualizations appear to provide a further display of the implications of our hypotheses, corroborating how regularization-induced mean-seeking is an integral component of current algorithms rather than a flawed artifact.

E.1. Behavior Model Pre-training Details

To produce Figure 3 in the main text and obtain the results in Appendix E, we pre-trained powerful autoregressive density models on the different MOOD datasets to act as a proxy for $\pi_{\mathcal{B}}$, which we denote $\pi'_{\mathcal{B}}$. We employ a 85/15 split to partition the trajectories into the training and validation datasets and employ early stopping based on the epoch achieving the highest validation log-likelihood. Our model can be conceptually split into to components: i) an observation encoder, outputting a latent representation ii) an action decoder, outputting a distribution for each action dimension by conditioning on the output of the observation encoder and on all previous action dimensions. Hence, to sample any action, the decoder must be queried $|A|$ times in an autoregressive fashion. However, we still compute the density of any particular action in a single forward pass at training time by basing our architecture on the seminal MADE model from Germain et al. (2015). The decoder output distribution we employ for each action dimension is a mixture of squashed Gaussians with ten independent components. Hence, for each action dimension, the autoregressive decoder outputs thirty values, representing the mean, log standard deviation, and weight logit for all mixture components. We found to obtain marginal gains with additional expressivity and that less-powerful models, such as simpler variational auto-encoders (Kingma & Welling, 2013) and affine Flows (Dinh et al., 2016), are unable to closely fit the distribution of behavior policies in the mixed-objective datasets. We refer to the shared code and Table 30 for further details and the employed hyper-parameters.