We propose methods for training world models on realistic offline datasets.

Take a photo for the full paper, talk and code.

Conservative World Models

Scott Jeen {srj38@cam.ac.uk}[1], Tom Bewley[2] & Jonathan M. Cullen[1]

[1] University of Cambridge, [2] University of Bristol

1 Background

Motivation. Zero-shot RL promises to provide agents capable of solving *any* task in an environment after offline pre-training, but existing methods require unrealistic, idealized datasets that cannot be expected for most real problems.

Forward-backward (FB) representations, the existing SOTA method, rely on *successor measures,* which are the expected discounted time spent in subsets of future states:

$$M^{\pi}(s_0, a_0, S_+) := \sum_{t=0}^{T-1} \gamma^t \Pr(s_{t+1} \in S_+ | (s_0, a_0), \pi), \ \forall \ S_+ \subset \mathcal{S}.$$

Together, a forward and backward model approximate successor measures for all policies

$$M^{\pi_z}(s_0, a_0, \mathrm{d}s_+) \approx F(s_0, a_0, z)^\top B(s_+) \rho(\mathrm{d}s_+), \ \forall \ s_+ \in \mathcal{S}.$$

They are trained with TD-learning according to:

$$\mathcal{L}_{\text{FB}} = \mathbb{E}_{(s_t, a_t, s_{t+1}, s_+) \sim \mathcal{D}, z \sim \mathcal{Z}} [\big(F(s_t, a_t, z)^\top B(s_+) - \gamma \bar{F}(s_{t+1}, \pi_z(s_{t+1}), z)^\top \bar{B}(s_+)\big)^2 - 2F(s_t, a_t, z)^\top B(s_{t+1})],$$

but the sampling of actions at the next state in this update causes value overestimation (Figure 1).

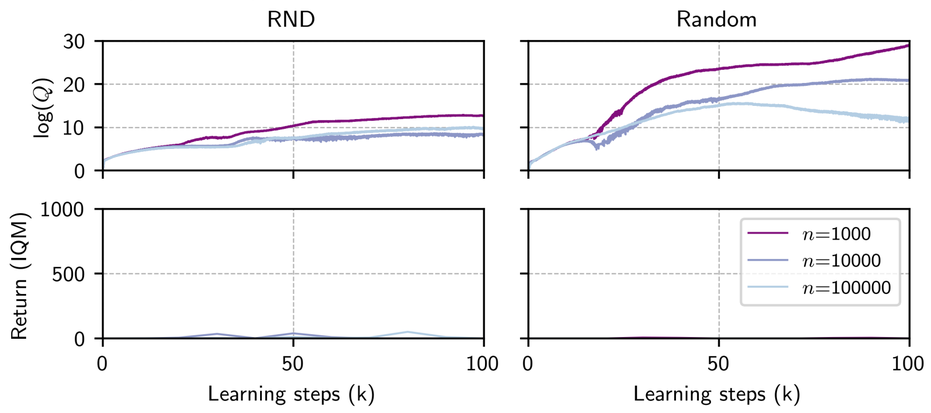

Figure 1: **FB value overestimation with respect to dataset size and quality.** Log Q values and IQM of rollout performance on all Maze tasks for datasets RND and Random

2 Conservative FB Representations

As a remedy, we propose *value-conservative forward-backward representations* (VC-FB) learned via:

$$\mathcal{L}_{\text{VC-FB}} = \alpha \cdot \Big(\mathbb{E}_{s \sim \mathcal{D}, z \sim \mathcal{Z}}[\max_a F(s, a, z)^\top z] - \mathbb{E}_{(s,a) \sim \mathcal{D}, z \sim \mathcal{Z}}[F(s, a, z)^\top z]\Big) + \mathcal{L}_{\text{FB}}.$$

And *measure-conservative forward-backward representations* (MC-FB) learned via:

$$\mathcal{L}_{\text{MC-FB}} = \alpha \cdot \Big(\mathbb{E}_{(s,s_+) \sim \mathcal{D}, z \sim \mathcal{Z}}[\max_a F(s, a, z)^\top B(s_+)] - \mathbb{E}_{(s,a,s_+) \sim \mathcal{D}}[F(s, a, z)^\top B(s_+)]\Big) + \mathcal{L}_{\text{FB}}.$$

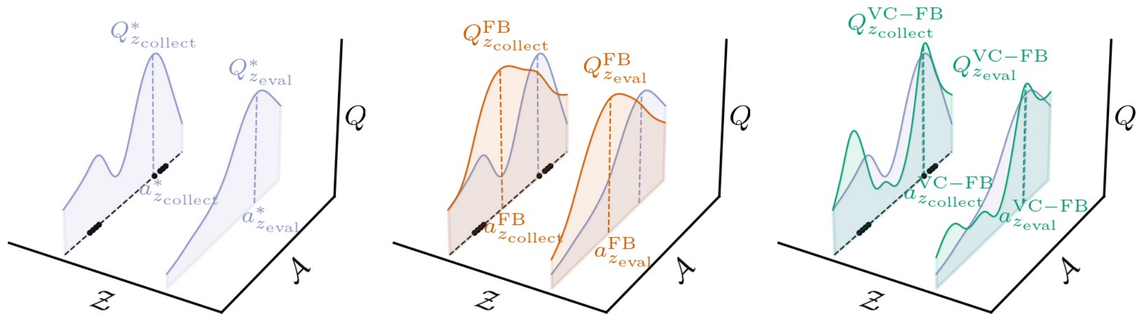

Figure 2: **FB's failure mode on sub-optimal datasets and VC-FB's resolution.** (*Left*) Zero-shot RL methods must generalize to any task in z-space. (*Middle*) FB overestimates the value of actions not in the dataset. (*Right*) VC-FB suppresses the value of actions not in the dataset.

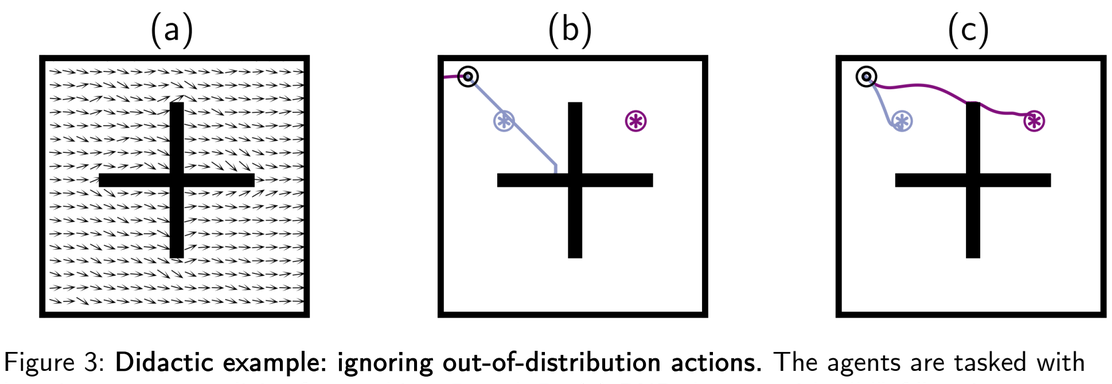

Figure 3: **Didactic example: ignoring out-of-distribution actions.** The agents are tasked with learning separate policies for reaching ⊛ and ⊛. (a) RND dataset with all "left" actions removed (b) Best FB rollout after 1 million steps. (c) Best VC-FB performance after 1 million learning steps.

3 Experiments

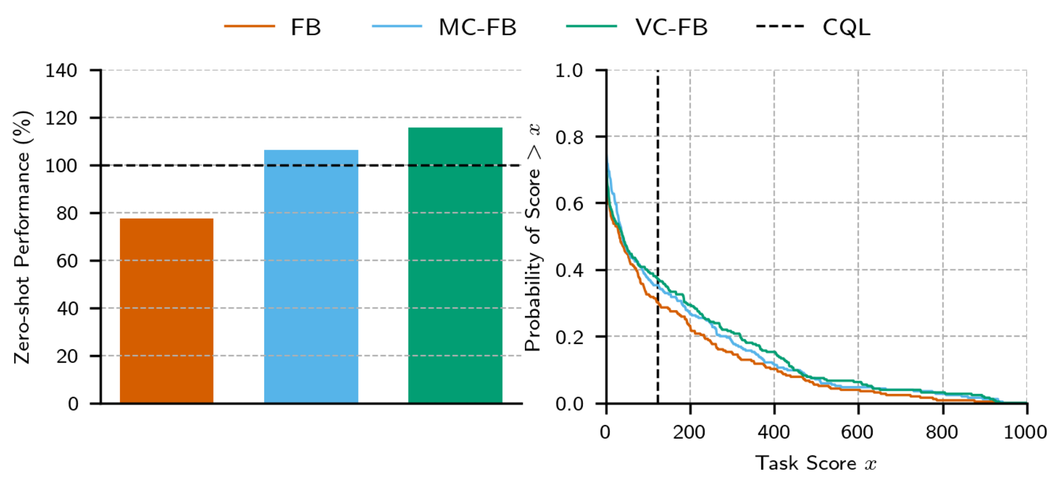

Figure 3: **Aggregate zero-shot performance.** (*Left*) IQM of task scores across datasets and domains, normalised against the performance of CQL, our baseline. (*Right*) Performance profiles showing the distribution of scores across all tasks and domains.

Setup. Each method was pre-trained on 3, 100k sub-sampled datasets from the ExORL benchmark (RND, DIAYN and Random); evaluated on 17 tasks across 4 domains DeepMind control suite domains.

Baselines. FB, CQL and Offline TD3.

Results.

- MC-FB and VC-FB stochastically dominate FB, achieving 150% and 137% its performance respectively;
- MC-FB and VC-FB outperform our CQL *despite not having access to task-specific reward labels and needing to fit policies for all tasks;*
- MC-FB and VC-FB perform no worse than FB on full datasets.

Table 1: **Performance on full RND dataset**. Aggregated IQM scores for all task with 95% confidence intervals, averaged across three seeds. Both VC-FB and MC-FB maintain the performance of FB on full datasets.

Domain	Task	FB	VC-FB	MC-FB
Walker	all tasks	639(616-661)	659(647-670)	651(632-672)
Quadruped	all tasks	656(638-674)	579(522-635)	635(628-642)
Point-mass Maze	all tasks	219(86-353)	287(117-457)	261(159-363)
Jaco	all tasks	39(29-50)	33(24-42)	34(18-51)
All	all tasks	361	**381**	**381**