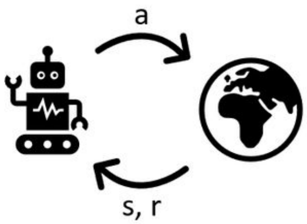
Consistent *Q*-Learning

Albert Wilcox, Atharva Mete, Chetan Reddy

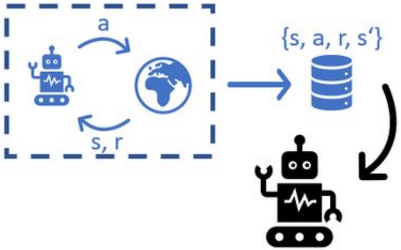
Motivation: Imitation Learning

Online Reinforcement Learning



- Requires active data collection
- Unsafe and expensive in real world

Imitation Learning/ Offline RL

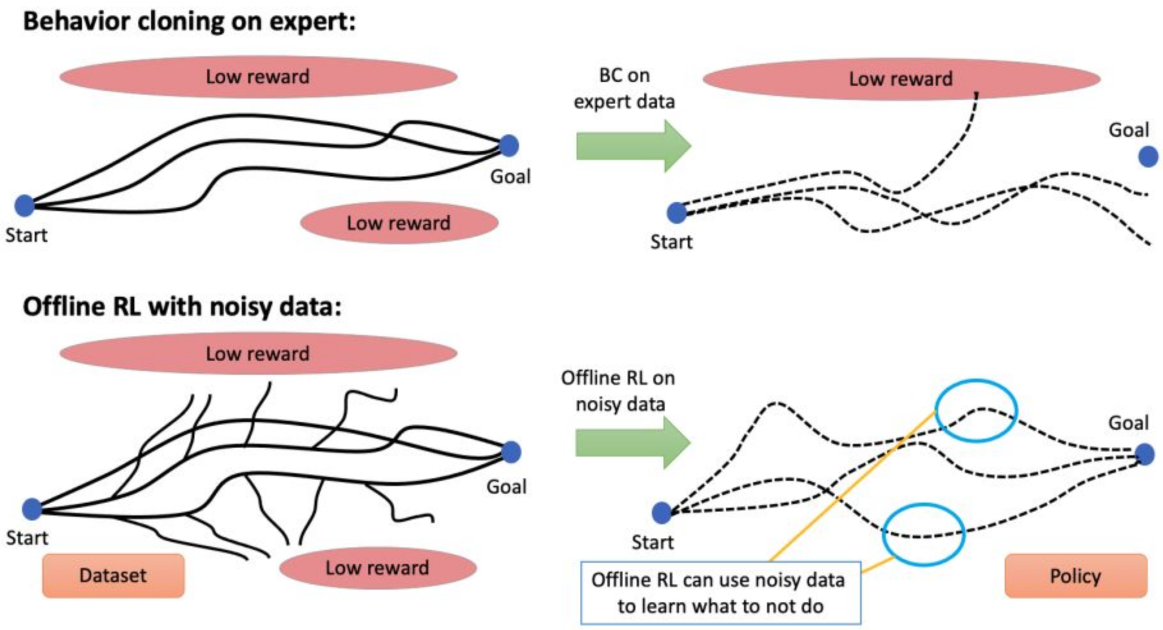


- Large-scale teleop data
- High learning signal per gradient update

1. Levine, et al. "Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems."

Motivation: Offline RL

- Robot learning at scale requires learning from diverse and highly **suboptimal** data



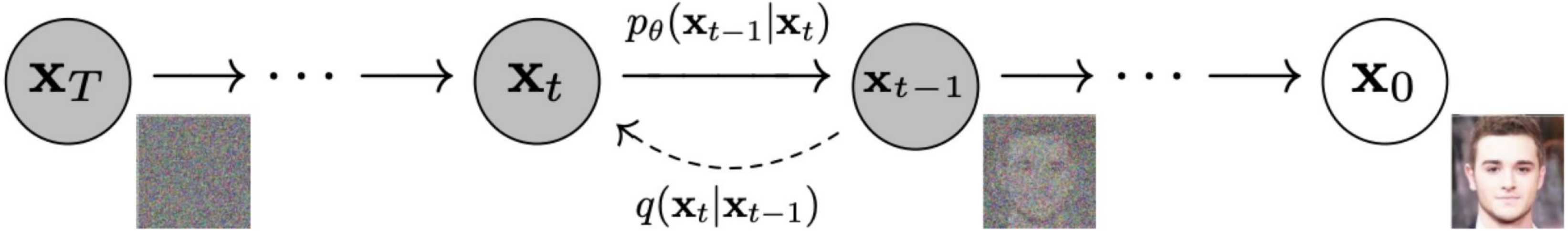
1. Kumar, Aviral, et al. "When should we prefer offline reinforcement learning over behavioral cloning?." *arXiv preprint arXiv:2204.05618* (2022).

Problem Statement

For a Markov decision process *(S, A, r(), p())*, given a fixed dataset *D* of reward-labeled suboptimal environment interaction tuples, the goal is to learn to maximize the expected discounted return under a learned policy π

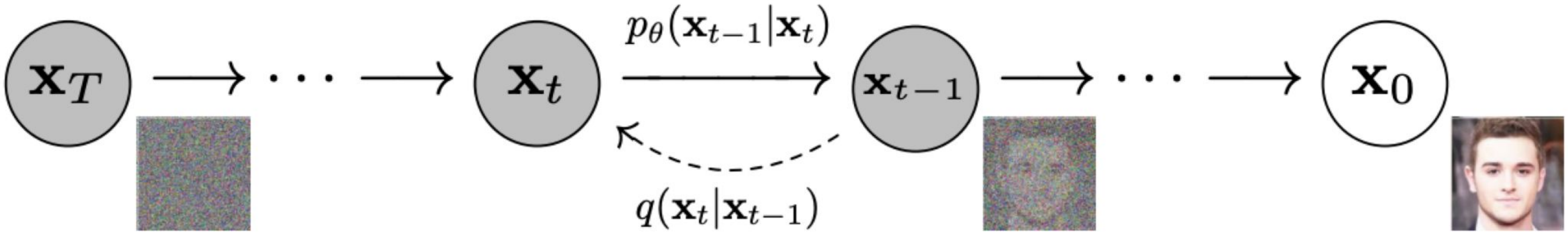
$$\text{argmax}_\pi \mathbb{E}_\pi \left[\sum_{i=0}^{\infty} \gamma^t r_{t+i} \right]$$

Background: Diffusion Models



- A class of **generative models** based on learning to ‘denoise’ to go from a prior distribution to a sample from the data distribution.
- Outperforms GANs & VAEs at modeling **complex, multi-modal distributions**.
- Connected to denoising score matching and Langevin dynamics.

Background: Diffusion Models



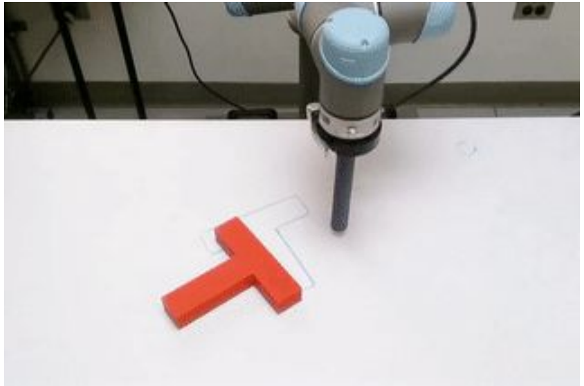
- Starting from image x_0, sample $\epsilon \sim \mathcal{N}(0,1), t \sim \text{Uniform}[1,T]$
- Compute noisy image $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$
- Compute denoising loss

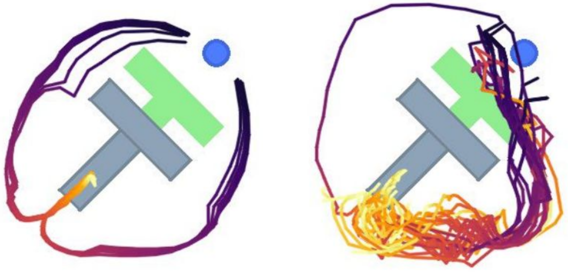
$$L(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[\left|\left|\epsilon - \epsilon_\theta(x_t, t)\right|\right|^2\right]$$

Background: Diffusion Policies

- Use diffusion model to predict receding horizon trajectories from images.
- Impressive results on wide range of complex multi-task settings
- Models diverse multimodal demonstrations well

- **Problem:** only BC ⇒ cannot exceed suboptimal demonstrations!





Diffusion Policy LSTM-GMM

1. Chi, Cheng, et al. "Diffusion policy: Visuomotor policy learning via action diffusion." arXiv preprint arXiv:2303.04137 (2023).

Related Works: Diffusion + RL

- Diffusion QL [1]: Offline RL + BC where the policy is represented as a DDPM
  - $$\pi = \arg\min_{\theta} \mathcal{L}(\theta) = \underbrace{\boxed{\mathcal{L}_{\mathrm{D}}(\theta)}}_{\text{BC term}} - \underbrace{\boxed{\alpha \mathbb{E}_{s\sim\mathcal{D}, a^0\sim\pi_\theta}\left[Q_\phi(s, a^0)\right]}}_{\text{Q target}}$$
  - During training it differentiates through the diffusion MC which is expensive
- Efficient Diffusion Policy [2]: approximates the diffusion chain in a way that compromises some multimodality properties, not scalable!
- Consistency policy [3] replaces the diffusion model with a consistency model
  - Not as expressive as a diffusion model for more difficult tasks!

Can we design an efficient algorithm without introducing detrimental approximations?

1. Wang, Z., Hunt, J. J., & Zhou, M. (2022). Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning. ICLR 2023.
2. Kang, B., Ma, X., Du, C., Pang, T., & Yan, S. (2023). Efficient Diffusion Policies for Offline Reinforcement Learning. arXiv [Cs.LG].
3. Ding, Z., & Jin, C. (2023). Consistency Models as a Rich and Efficient Policy Class for Reinforcement Learning. arXiv [Cs.LG]. Retrieved from http://arxiv.org/abs/2309.16984

Key Idea

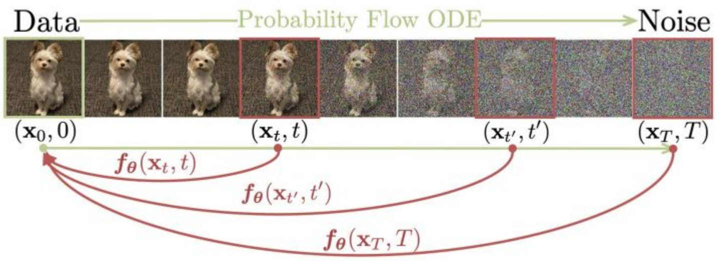
These algorithms struggle because they can only evaluate the denoising network based on **final, noiseless, actions** at the end of the diffusion denoising process

On the other hands, diffusion models learn by computing meaningful gradients throughout the diffusion chain

We need a method to evaluate noisy actions throughout the diffusion chain so that we can improve the denoising process throughout

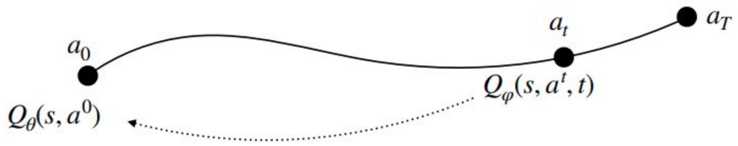
Consistency Models

- Consistency models [1] learn to predict denoised samples from anywhere in the denoising chain
- Decent quality single step samples, also supports iteratively refining for better quality if desired



Our Idea:

- Learn a consistent critic, which can accurately predict Q-values at any point in the diffusion chain
- Use the consistent critic to update the policy without differentiating all the way through the diffusion chain



1. Song, Y., Dhariwal, P., Chen, M., & Sutskever, I. (2023). Consistency models. ICML 2023

Consistent Q Learning: IQL Critic

- **Problem:** diffusion models are expensive to query, and a typical Bellman backup-style critic update requires querying the policy to get critic targets
- **Solution:** IQL [1] learns a critic without querying the current policy by maximizing an expectile over the dataset
  - Value function maximizes expectile over dataset, where $L_2^\tau(u) = |\tau - \mathbb{1}(u < 0)|u^2$

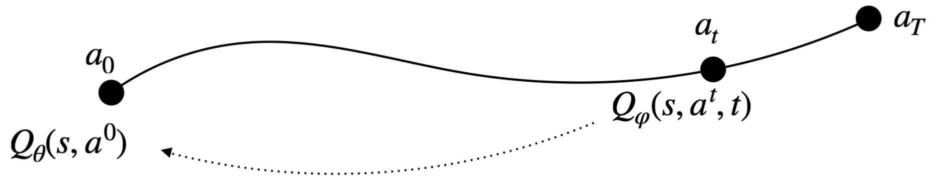
$$L_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}[L_2^\tau(Q_{\hat{\theta}}(s,a) - V_\psi(s))]$$

  - Update critic using the value function

$$L_Q(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}[(r(s,a) + \gamma V_\psi(s') - Q_\theta(s,a))^2]$$

1. Kostrikov, I., Nair, A., & Levine, S. (2021). Offline Reinforcement Learning with Implicit Q-Learning.

Consistent Q Learning: Consistent Critic



- Parameterize Q_φ using a similar trick to [1]

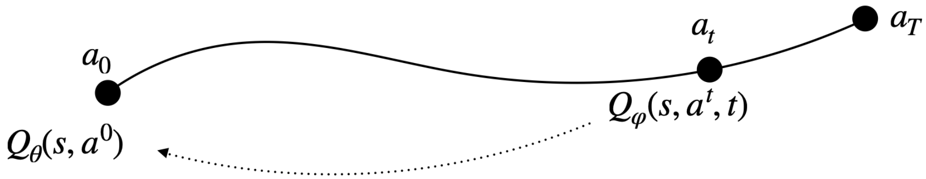
$$Q_\varphi(s, a^t, t) = c_{\text{skip}}(t) Q_\theta(s, a^t) + c_{\text{out}}(t) F_\varphi(s, a^t, t)$$

- Sample t, compute a^t, take one denoising step to compute a^{t-1} and train according to consistency loss

$$\mathscr{L}(\varphi) = \text{MSE}\left(Q_\varphi(s, a^{t-1}, t-1), Q_\varphi(s, a^t, t) \right)$$

1. Song, Y., Dhariwal, P., Chen, M., & Sutskever, I. (2023). Consistency models. ICML 2023

Consistent Q Learning: Policy Update



- Compute noisy action a^t and denoise by one step to a^{t-1}
- Use the consistent critic to compute an advantage weighted denoising update

$$\mathcal{L}(\theta) = \mathbb{E}\left[\exp(\beta(Q_\theta(s, a_{t-1}, t-1) - V_\psi(s)))\left|\left|\epsilon - \epsilon_\theta(a_t, t)\right|\right|^2\right]$$

Advantage weighting

Action denoising loss

Experiments

We perform experiments to test following hypotheses:

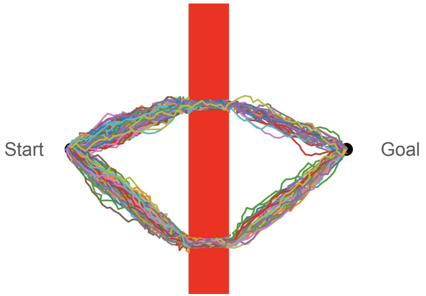
1. CoQL is comparable with or improves upon baseline results
   - D4RL

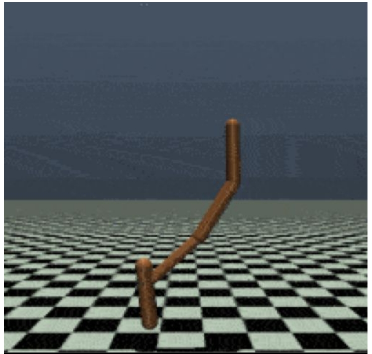
2. Consistent critic provides more accurate Q-value estimations at noisy actions
   - Using original critic (not trained on noisy actions) ablation

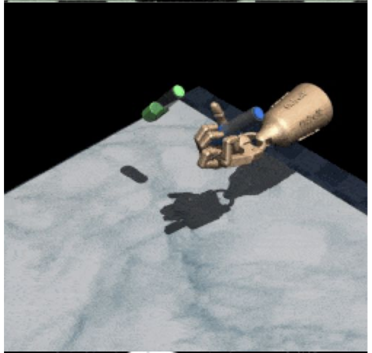
3. CoQL performs better than baselines on high-dimensional tasks, as we don’t have to approximate the diffusion process
   - Dexterous tasks

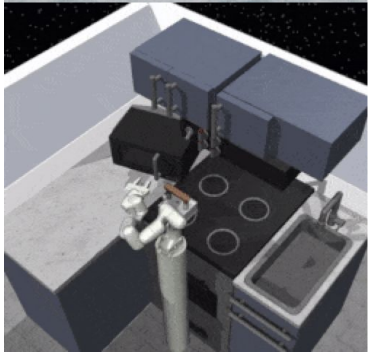
Domains

- D4RL [1] is a widely used offline RL benchmark
- Locomotion environments
  - Hopper, Walker2D, Half-Cheetah
  - medium, medium-replay, medium-expert
- Adroit
  - Repositioning Pen
  - High-dimensional
- Kitchen
  - Very multimodal
  - Requires trajectory stitching to solve
- Navigation:
  - Sequence of actions
  - Opt, noisy, slow, slow-noisy









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

Results

- D4RL:

Environment	DQL	EDP	No Consistent	CoQL (tune)	CoQL (best)
halfcheetah-medium-v2	0.508 ± 0.005	0.521	0.545 ± 0.003	$\mathbf{0.608 \pm 0.006}$	0.511 ± 0.001
hopper-medium-v2	$\mathbf{0.822 \pm 0.030}$	0.819	0.690 ± 0.032	0.685 ± 0.095	0.685 ± 0.095
walker2d-medium-v2	$\mathbf{0.871 \pm 0.006}$	0.869	0.432 ± 0.198	0.863 ± 0.014	0.822 ± 0.013
halfcheetah-medium-replay-v2	0.474 ± 0.001	0.494	0.512 ± 0.004	$\mathbf{0.532 \pm 0.013}$	0.467 ± 0.016
hopper-medium-replay-v2	1.004 ± 0.003	1.010	$\mathbf{1.025 \pm 0.003}$	0.898 ± 0.135	0.898 ± 0.135
walker2d-medium-replay-v2	0.871 ± 0.071	**0.949**	0.902 ± 0.002	0.833 ± 0.051	0.803 ± 0.018
halfcheetah-medium-expert-v2	0.953 ± 0.011	0.955	$\mathbf{0.958 \pm 0.024}$	0.734 ± 0.202	0.390 ± 0.089
hopper-medium-expert-v2	$\mathbf{1.061 \pm 0.067}$	0.974	0.177 ± 0.052	0.792 ± 0.391	0.694 ± 0.189
walker2d-medium-expert-v2	1.099 ± 0.001	**1.102**	1.012 ± 0.082	1.029 ± 0.017	1.029 ± 0.017
Average	0.853	**0.869**	0.717	0.793	0.708
pen-human-v1	0.590 ± 0.106	**0.727**	0.712 ± 0.156	0.657 ± 0.213	0.657 ± 0.213
pen-cloned-v1	0.452 ± 0.130	**0.700**	0.558 ± 0.057	0.66 ± 0.163	0.611 ± 0.134
Average	0.521	**0.714**	0.635	0.659	0.634
kitchen-complete-v0	$\mathbf{0.775 \pm 0.088}$	0.755	0.742 ± 0.155	0.725 ± 0.235	0.725 ± 0.235
kitchen-partial-v0	0.528 ± 0.078	0.528	0.625 ± 0.074	$\mathbf{0.717 \pm 0.024}$	$\mathbf{0.717 \pm 0.024}$
kitchen-mixed-v0	0.519 ± 0.047	0.608	$\mathbf{0.692 \pm 0.042}$	$\mathbf{0.692 \pm 0.012}$	0.625 ± 0.071
Average	0.607	0.630	0.686	**0.711**	0.689

Results

- Navigation:
  - Success rate

Environment	BC	IDQL	No Consistent	CoQL
nav-ms-opt	$\mathbf{0.933 \pm 0.047}$	0.8 ± 0.082	$\mathbf{0.933 \pm 0.047}$	$\mathbf{0.933 \pm 0.094}$
nav-ms-slow-noisy	0.467 ± 0.047	$\mathbf{0.767 \pm 0.047}$	0.6 ± 0.082	0.6 ± 0.0
nav-ms-slow	0.767 ± 0.047	$\mathbf{0.967 \pm 0.047}$	0.9 ± 0.082	0.833 ± 0.125
nav-ms-noisy	0.3 ± 0.082	$\mathbf{0.667 \pm 0.047}$	0.6 ± 0.082	0.633 ± 0.094
Average	0.617	0.800	0.758	0.750

  - Average Reward

Environment	BC	IDQL	No Consistent	CoQL
nav-ms-opt	-57.833 ± 2.151	-63.433 ± 5.898	$\mathbf{-53.8 \pm 3.395}$	-65.867 ± 14.751
nav-ms-slow-noisy	-203.733 ± 18.184	$\mathbf{-181.533 \pm 7.376}$	-186.433 ± 4.84	-199.8 ± 9.819
nav-ms-slow	-150.4 ± 11.064	$\mathbf{-56.633 \pm 2.53}$	-63.7 ± 6.255	-68.5 ± 8.702
nav-ms-noisy	-110.233 ± 5.473	$\mathbf{-86.367 \pm 4.203}$	-103.3 ± 13.003	-89.533 ± 6.884
Average	-130.5	-97.0	-101.8	-105.9

Conclusion

- In this project we present consistent Q learning
- The proposed method shows some initial progress but largely struggles to outperform comparison algorithms
- Results suggest that the original critic also gives “good” value estimates on noisy action, we suppose this is due to the fact that adding noise smooths out the overall gradient landscape

- In the future, we plan to
  - Explore other policy improvement formulations that may work better than the advantage weighting objective
  - Experiment with more complicated environments where the proposed method may scale better

Questions?