Supported Policy Optimization for Offline Reinforcement Learning

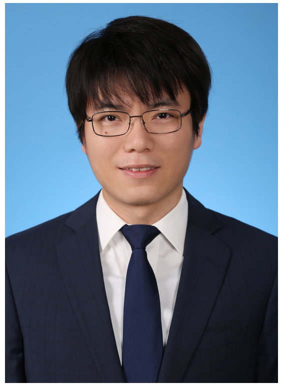
Jialong Wu[1], Haixu Wu[1], Zihan Qiu[2], Jianmin Wang[1], Mingsheng Long[1]✉
[1]School of Software, BNRist, Tsinghua University, China
[2]Institute for Interdisciplinary Information Sciences, Tsinghua University, China
`{wujialong0229,qzh11628}@gmail.com, whx20@mails.tsinghua.edu.cn`
`{jimwang,mingsheng}@tsinghua.edu.cn`


Jialong Wu

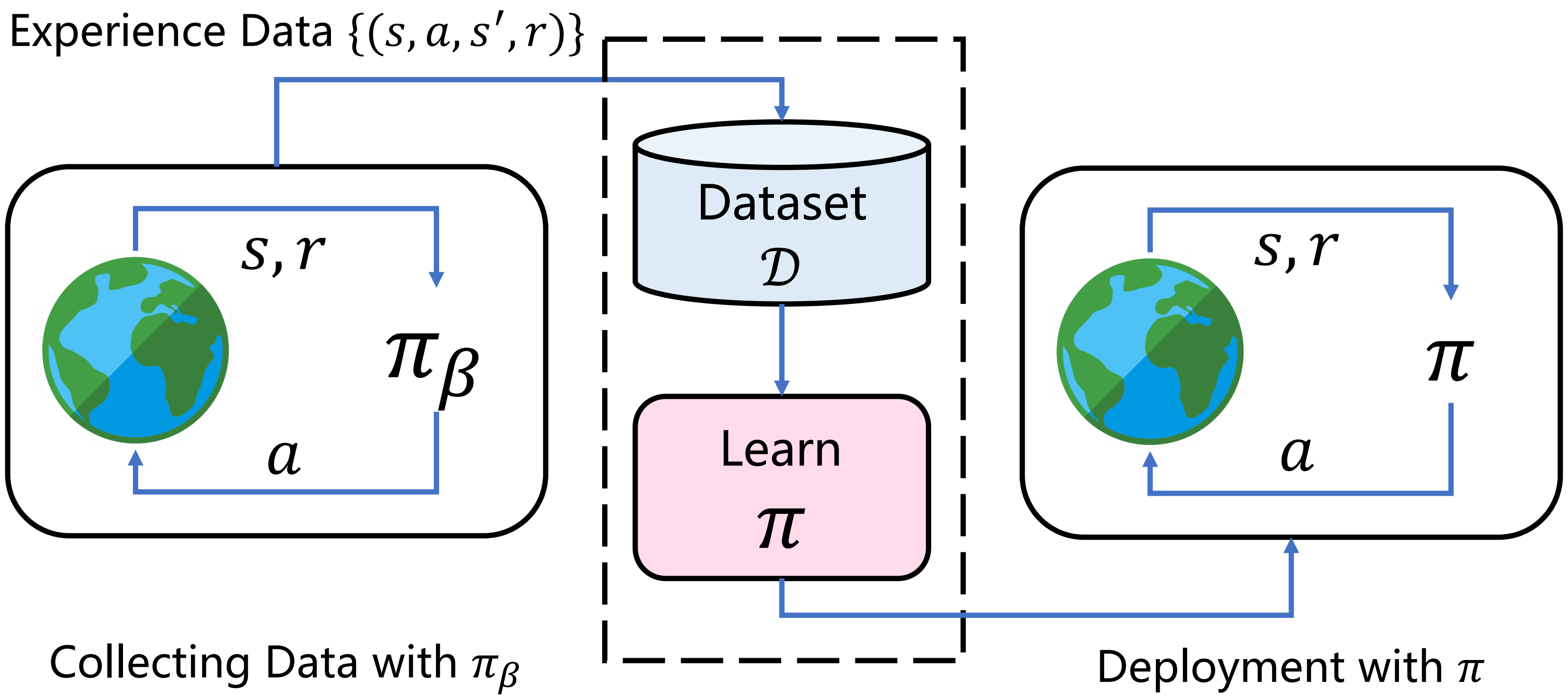

Haixu Wu


Zihan Qiu


Jianmin Wang

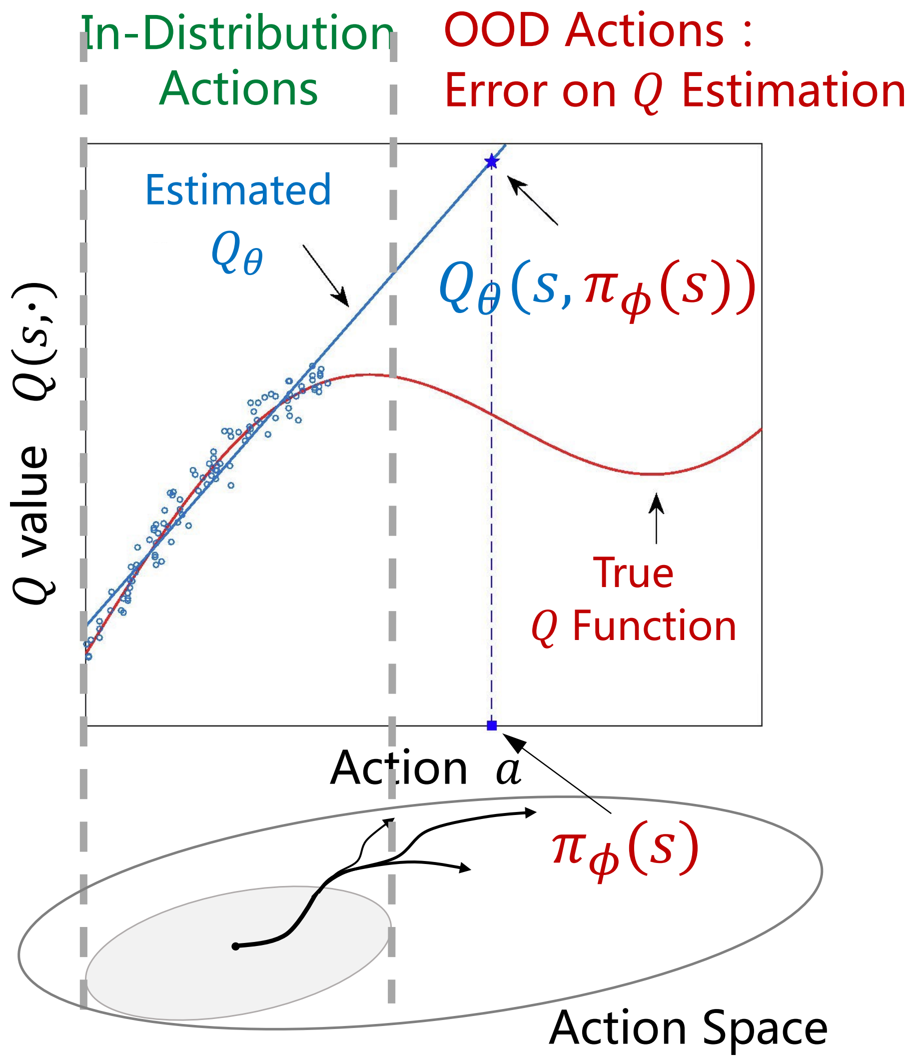

Mingsheng Long

Offline Reinforcement Learning



Eliminating the need to expensive or risky interactions with the live environment in practical scenarios

Extrapolation Error in Offline RL

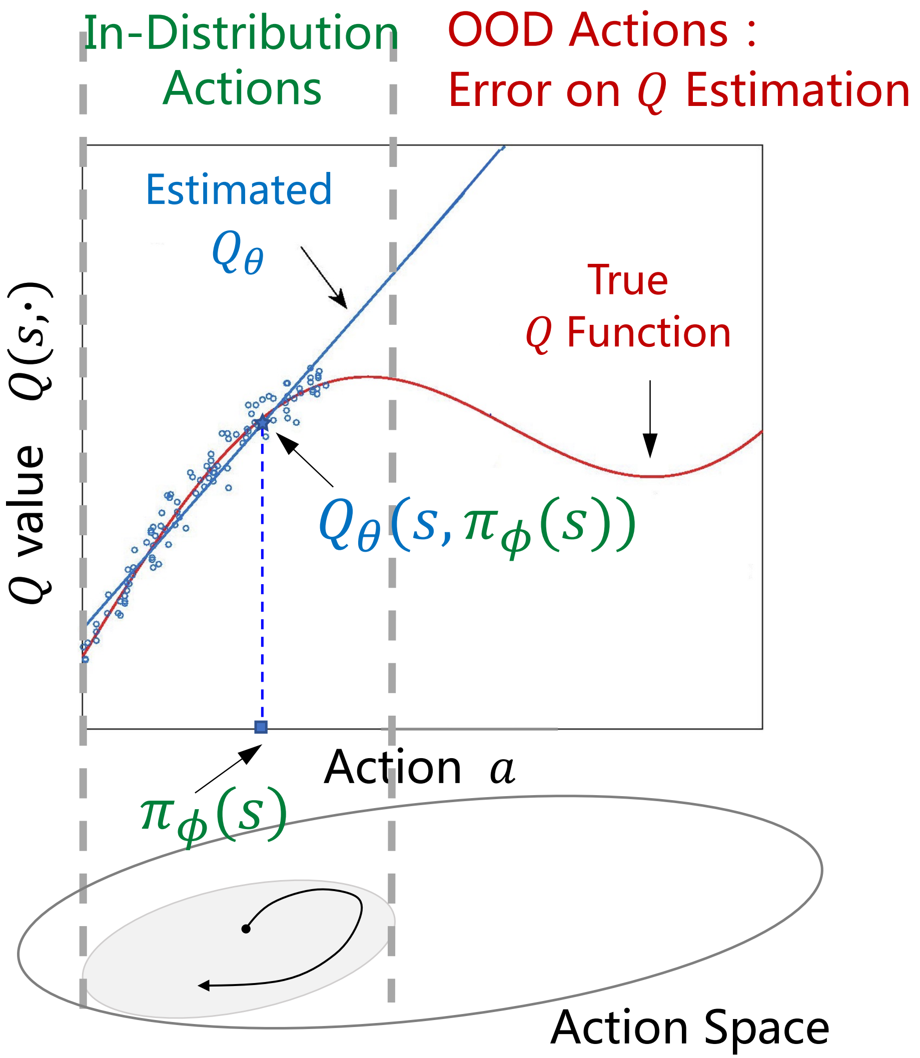


$$J_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{D}}\left[-\boxed{Q_\theta\left(s, \pi_\phi(s)\right)}\right]$$

Extrapolation Error of Q Estimation

- Misleading policy gradient
- Error propagation through Bellman backups

Support Constraint in Offline RL



$$\min_{\phi} \quad J_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{D}}\left[-Q_\theta\left(s, \pi_\phi(s)\right)\right]$$

$$\text{s.t.} \quad \boxed{\pi_\phi(s) \in \{a : \pi_\beta(a \mid s) > \epsilon\}} \quad \forall s$$

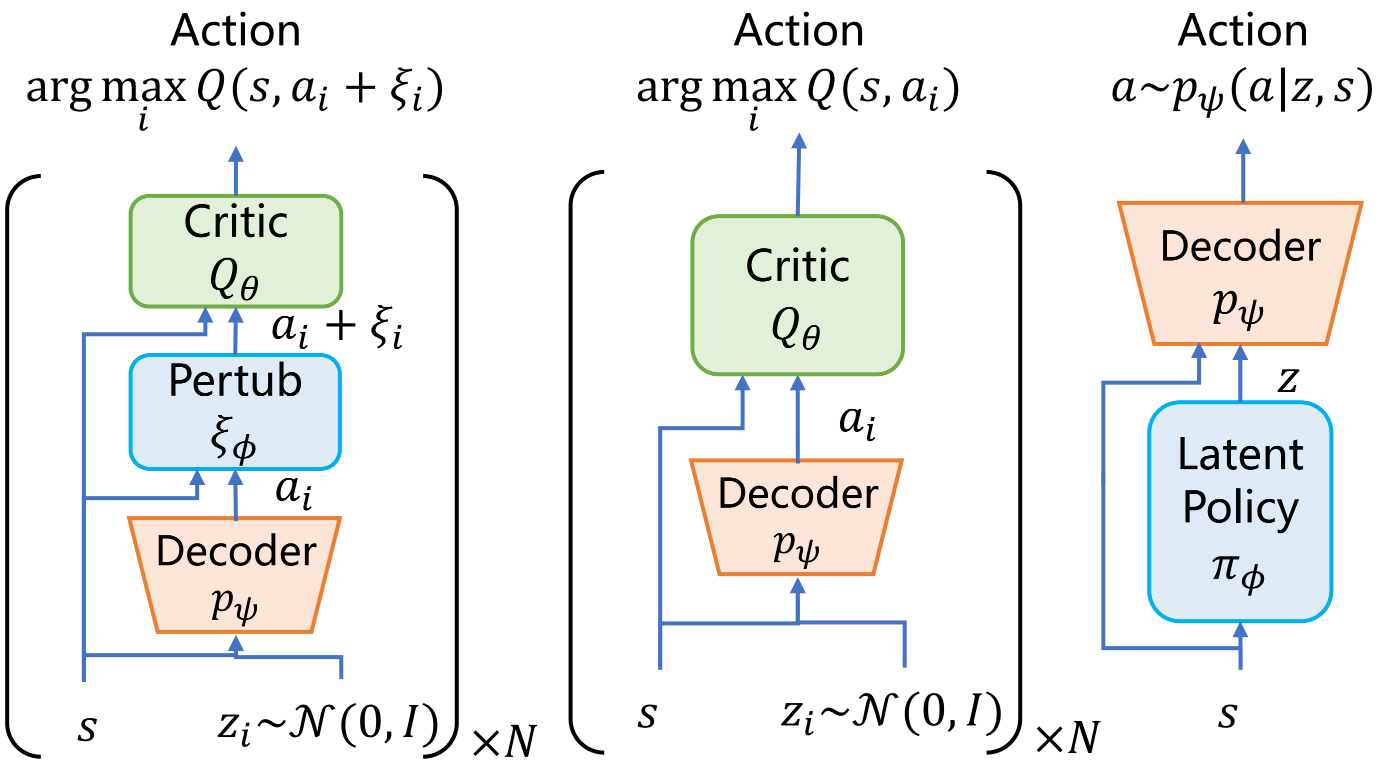
Support Constraint

- Tradeoff between optimality and extrapolation error [Kumar et al. NeurIPS 19]

Parameterization vs Regularization

Support Constraint via Parameterization

- Policy coupled to **generative models**
- Pros
  - Direct constraint
- Cons
  - Extra inference cost
  - Implementation difficulty
  - Complicates transfer of design techniques



BCQ [Fujimoto et al., ICML 2019]

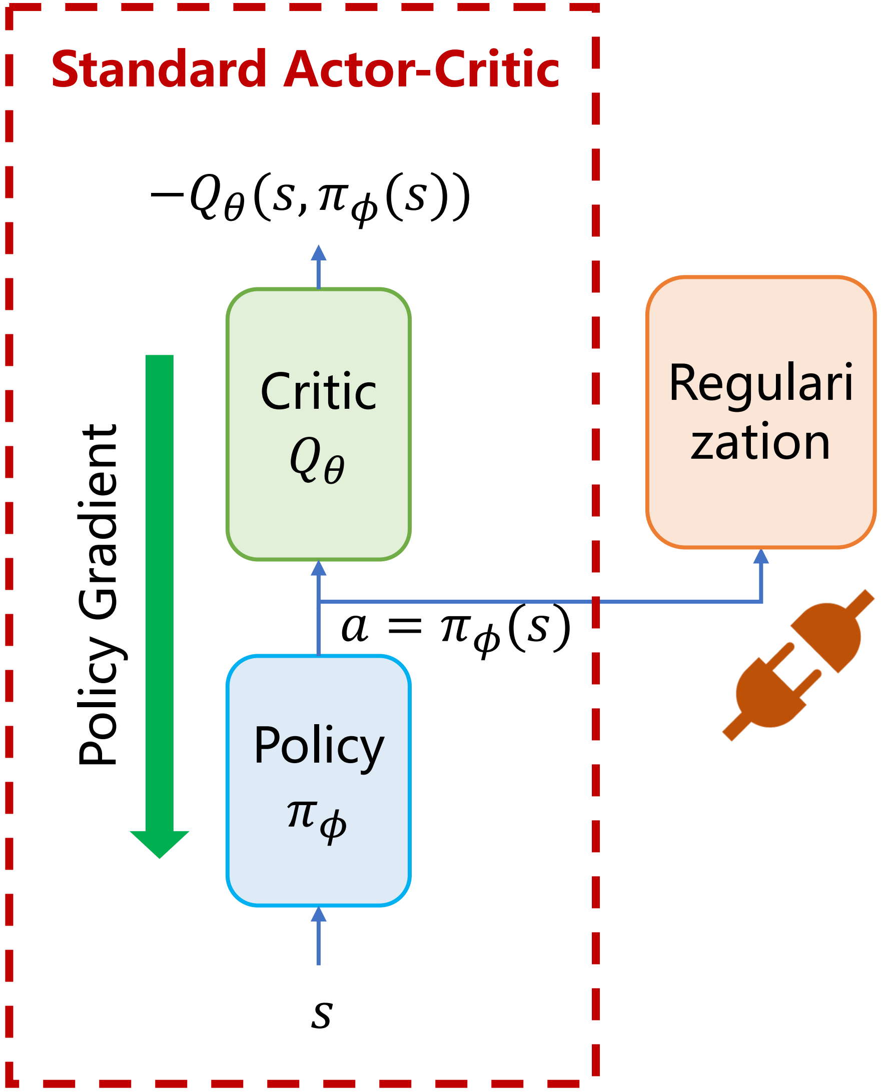
EMaQ [Ghasemipour et al., ICML 2021]

PLAS [Zhou et al., CoRL 2020]

Parameterization vs Regularization

Support Constraint via Regularization

- Penalize **divergence** between π and π_β
  - MMD [Kumar et al., 2019]
  - Wasserstein distance [Wu et al., 2019]
  - Behavior cloning term [Fujimoto & Gu, 2021]
- **Pros**
  - Pluggable design
- **Cons**
  - Mismatch the inherent density-based definition of support constraint



Parameterization vs Regularization

Support Constraint via Regularization

- Penalize **divergence** between π and π_β
  - MMD [Kumar et al., 2019]
  - Wasserstein distance [Wu et al., 2019]
  - Behavior cloning term [Fujimoto & Gu, 2021]
- **Pros**
  - Pluggable design
- **Cons**
  - Mismatch the inherent density-based definition of support constraint

Our Goal:

- A **pluggable** offline RL method that also **directly** meets the support constraint

Support Constraint via Behavior Density

$$\max_{\phi} \mathbb{E}_{s\sim\mathcal{D}}\left[Q_\theta(s,\pi_\phi(s))\right]$$
$$\text{s.t.} \min_{s} \log \pi_\beta(\pi_\phi(s)|s) > \hat{\epsilon}$$

Policy optimization
with behavior density as constraint

Support Constraint via Behavior Density

$$\max_{\phi} \mathbb{E}_{s\sim\mathcal{D}} \left[Q_\theta(s, \pi_\phi(s))\right]$$

$$\text{s.t.} \min_{s} \log \pi_\beta(\pi_\phi(s)|s) > \hat{\epsilon}$$

Policy optimization
with behavior density as constraint

$$\max_{\phi} \mathbb{E}_{s\sim\mathcal{D}} \left[Q_\theta(s, \pi_\phi(s))\right]$$

$$\text{s.t.} \mathbb{E}_{s\sim\mathcal{D}} \left[\log \pi_\beta(\pi_\phi(s)|s)\right] > \hat{\epsilon}$$

**Heuristic approximation
widely adopted by RL literatures**

Support Constraint via Behavior Density

$$\max_{\phi} \mathbb{E}_{s\sim\mathcal{D}}\left[Q_\theta(s, \pi_\phi(s))\right]$$

$$\text{s.t.} \min_{s} \log \pi_\beta(\pi_\phi(s)|s) > \hat{\epsilon}$$

Policy optimization
with behavior density as constraint

$$\max_{\phi} \mathbb{E}_{s\sim\mathcal{D}}\left[Q_\theta(s, \pi_\phi(s))\right]$$

$$\text{s.t.} \mathbb{E}_{s\sim\mathcal{D}}\left[\log \pi_\beta(\pi_\phi(s)|s)\right] > \hat{\epsilon}$$

Heuristic approximation
widely adopted by RL literatures

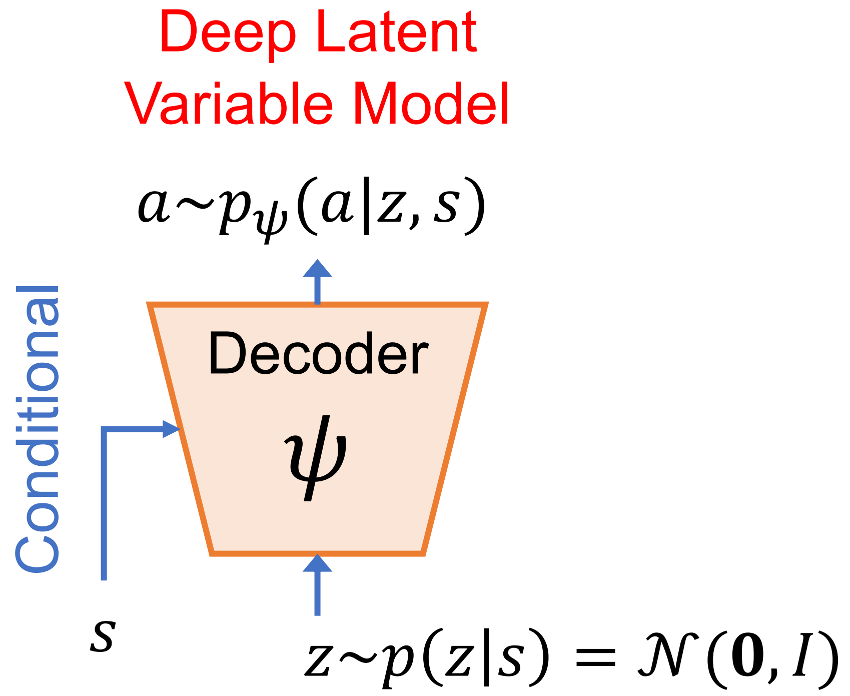
$$J_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{D}}\left[-Q_\theta\left(s, \pi_\phi(s)\right) - \lambda \log \pi_\beta\left(\pi_\phi(s)|s\right)\right]$$

**Policy
learning objective**

Explicit Estimation of Behavior Density

Conditional Variational Auto-Encoder (CVAE)

$$\pi_\beta(a|s) = p_\psi(a|s) = \int p_\psi(a|z,s)p(z|s)\mathrm{d}z$$



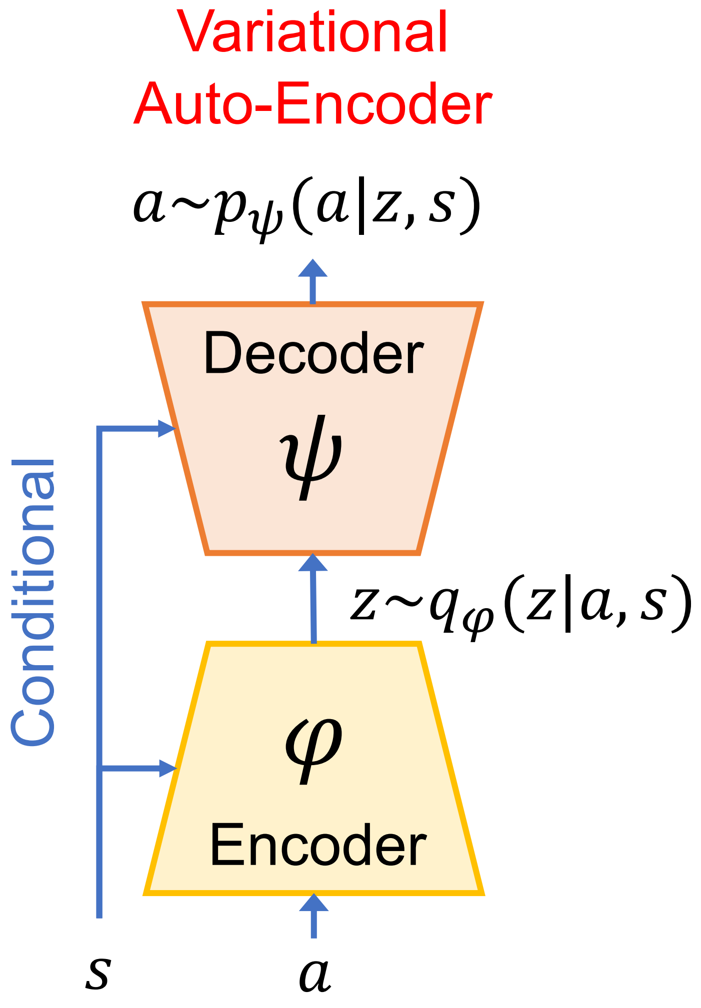
Explicit Estimation of Behavior Density

Conditional Variational Auto-Encoder (CVAE)

$$\pi_\beta(a|s) = p_\psi(a|s) = \int p_\psi(a|z,s)p(z|s)\mathrm{d}z$$

Optimization with evidence lower bound (ELBO)

$$\begin{aligned}\log p_\psi(a|s) &\geq \mathbb{E}_{q_\varphi(z|a,s)}\left[\log \frac{p_\psi(a,z|s)}{q_\varphi(z|a,s)}\right] \\ &= \boxed{\mathbb{E}_{q_\varphi(z|a,s)}\left[\log p_\psi(a|z,s)\right] - \mathrm{KL}\left[q_\varphi(z|a,s)\|p(z|s)\right]} \\ &\overset{\text{def}}{=} -\mathcal{L}_{\text{ELBO}}(s,a;\varphi,\psi).\end{aligned}$$

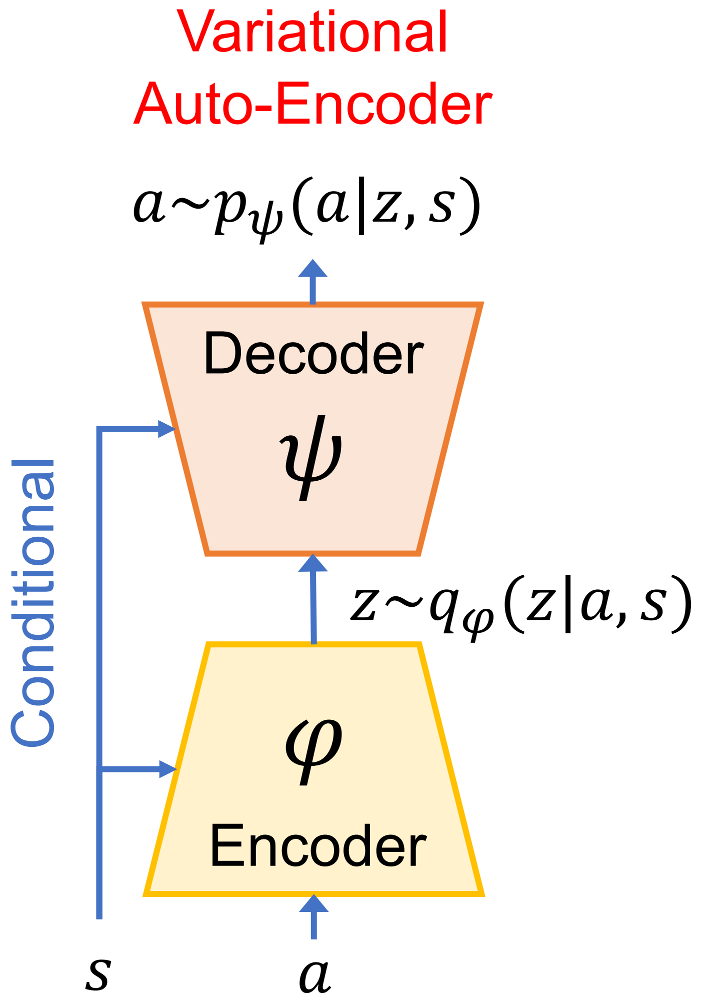


Explicit Estimation of Behavior Density

Density estimation with importance sampling

[Rezende et al., ICML 2014]

$$
\begin{aligned}
\log p_\psi(a|s) &= \log \mathbb{E}_{q_\varphi(z|a,s)}\left[\frac{p_\psi(a,z|s)}{q_\varphi(z|a,s)}\right] \\
&\approx \boxed{\mathbb{E}_{z^{(l)}\sim q_\varphi(z|a,s)}\left[\log \frac{1}{L}\sum_{l=1}^{L}\frac{p_\psi(a,z^{(l)}|s)}{q_\varphi(z^{(l)}|a,s)}\right]} \\
&\stackrel{\text{def}}{=} \widehat{\log \pi_\beta}(a|s;\varphi,\psi,L).
\end{aligned}
$$



Explicit Estimation of Behavior Density

Density estimation with importance sampling

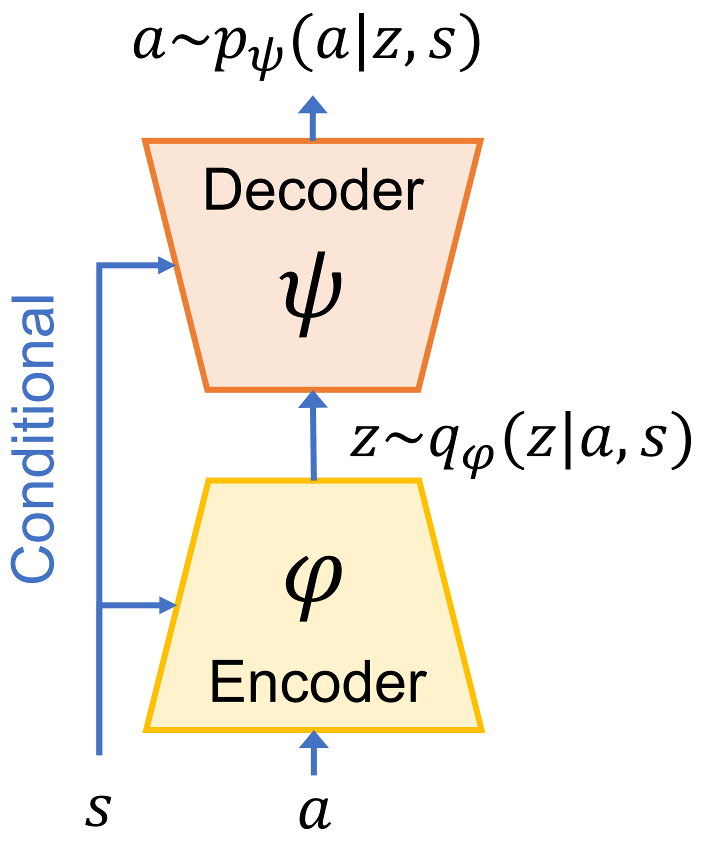
[Rezende et al., ICML 2014]

$$\log p_\psi(a|s) = \log \mathbb{E}_{q_\varphi(z|a,s)}\left[\frac{p_\psi(a,z|s)}{q_\varphi(z|a,s)}\right]$$

$$\approx \boxed{\mathbb{E}_{z^{(l)}\sim q_\varphi(z|a,s)}\left[\log \frac{1}{L}\sum_{l=1}^{L}\frac{p_\psi(a,z^{(l)}|s)}{q_\varphi(z^{(l)}|a,s)}\right]}$$

$$\stackrel{\text{def}}{=} \widehat{\log \pi_\beta}(a|s;\varphi,\psi,L).$$

Variational Auto-Encoder



Policy learning objective with density estimator

$$J_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{D}}\left[-Q_\theta\left(s,\pi_\phi(s)\right) - \lambda\widehat{\log \pi_\beta}(\pi_\phi(s)|s;\varphi,\psi,L)\right]$$

Supported Policy Optimization

Algorithm 1 Supported Policy Optimization (SPOT)

Input: Dataset $\mathcal{D} = \{(s, a, r, s')\}$
// VAE Training
Initialize VAE with parameters ψ and φ
for $t = 1$ **to** T_1 **do**
  Sample minibatch of transitions $(s, a) \sim \mathcal{D}$
  Update ψ, φ minimizing $\mathcal{L}_{\mathrm{ELBO}}(s, a; \varphi, \psi)$ in Eq. (7)
end for

(1). Density Estimation with VAE

// Policy Training
Initialize the policy network π_ϕ, critic network Q_θ and target network $Q_{\bar{\theta}}$ with $\bar{\theta} \leftarrow \theta$
for $t = 1$ **to** T_2 **do**
  Sample minibatch of transitions $(s, a, r, s') \sim \mathcal{D}$
  Update θ minimizing $J_Q(\theta)$ in Eq. (1)
  Update ϕ minimizing $J_\pi(\phi)$ in Eq. (9)
  Update target network: $\bar{\theta} \leftarrow \tau\theta + (1 - \tau)\bar{\theta}$
end for

(2). Actor-Critic with Plugged Regularization

Supported Policy Optimization

Algorithm 1 Supported Policy Optimization (SPOT)

Input: Dataset $\mathcal{D} = \{(s, a, r, s')\}$
// VAE Training
Initialize VAE with parameters ψ and φ
for $t = 1$ **to** T_1 **do**
  Sample minibatch of transitions $(s, a) \sim \mathcal{D}$
  Update ψ, φ minimizing $\mathcal{L}_{\text{ELBO}}(s, a; \varphi, \psi)$ in Eq. (7)
end for
// Policy Training
Initialize the policy network π_ϕ, critic network Q_θ and targ
for $t = 1$ **to** T_2 **do**
  Sample minibatch of transitions $(s, a, r, s') \sim \mathcal{D}$
  Update θ minimizing $J_Q(\theta)$ in Eq. (1)
  Update ϕ minimizing $J_\pi(\phi)$ in Eq. (9)
  Update target network: $\bar{\theta} \leftarrow \tau\theta + (1 - \tau)\bar{\theta}$
end for

Practical Implementation

- Base algorithm: TD3
- Q normalization following TD3+BC [Fujimoto & Gu, NeurIPS 2021]
- Simpler density estimator with $L = 1$ (ELBO estimator)

Experimental Evaluation on D4RL-Gym-MuJoCo

State-of-the-art performance on locomotion tasks

Table 2: Performance of SPOT and prior methods on Gym-MuJoCo tasks. m = medium, m-r = medium-replay, m-e = medium-expert. For baselines, we report numbers directly from the IQL paper [25], which provides a unified comparison for "-v2" datasets. For SPOT, we report the mean and standard deviation for 10 seeds.

	BC	AWAC	DT	Onestep	TD3+BC	CQL	IQL	SPOT (Ours)
HalfCheetah-m-e-v2	55.2	42.8	86.8	**93.4**	90.7	91.6	86.7	86.9±4.3
Hopper-m-e-v2	52.5	55.8	**107.6**	103.3	98.0	105.4	91.5	99.3±7.1
Walker-m-e-v2	107.5	74.5	108.1	**113.0**	110.1	108.8	109.6	112.0±0.5
HalfCheetah-m-v2	42.6	43.5	42.6	48.4	48.3	44.0	47.4	**58.4**±1.0
Hopper-m-v2	52.9	57.0	67.6	59.6	59.3	58.5	66.2	**86.0**±8.7
Walker-m-v2	75.3	72.4	74.0	81.8	83.7	72.5	78.3	**86.4**±2.7
HalfCheetah-m-r-v2	36.6	40.5	36.6	38.1	44.6	45.5	44.2	**52.2**±1.2
Hopper-m-r-v2	18.1	37.2	82.7	97.5	60.9	95.0	94.7	**100.2**±1.9
Walker-m-r-v2	26.0	27.0	66.6	49.5	81.8	77.2	73.8	**91.6**±2.8
Gym-MuJoCo total	466.7	450.7	672.6	684.6	677.4	698.5	692.4	**773.0**±30.2

Experimental Evaluation on D4RL-AntMaze

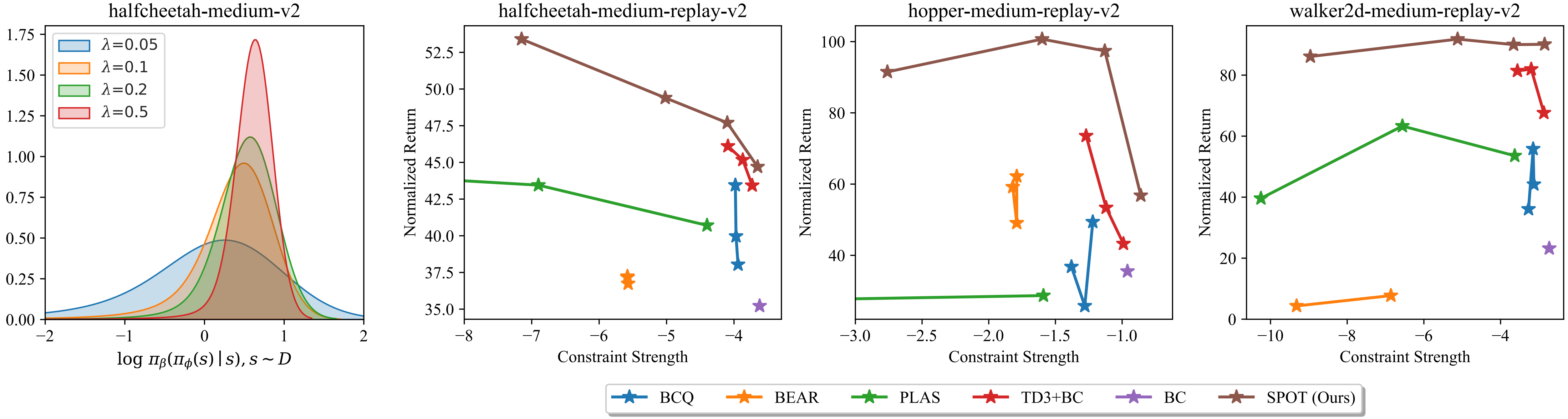
Strong performance with a simple pluggable design

Table 3: Performance of SPOT and prior methods on AntMaze tasks. For baselines, we obtain the results using author-provided implementations on "-v2" datasets. For BCQ and BEAR, we report numbers from D4RL paper [6]. For SPOT, we report the mean and standard deviation for 10 seeds.

	BCQ	BEAR	BC	DT	TD3+BC	PLAS	CQL	IQL	SPOT (Ours)
umaze-v2	78.9	73.0	49.2	54.2±4.1	73.0±34.0	62.0±16.7	82.6±5.7	89.6±4.2	**93.5**±2.4
umaze-diverse-v2	55.0	61.0	41.8	41.2±11.4	47.0±7.3	45.4±7.9	10.2±6.7	**65.6**±8.3	40.7±5.1
medium-play-v2	0.0	0.0	0.4	0.0±0.0	0.0±0.0	31.4±21.5	59.0±1.6	**76.4**±2.7	74.7±4.6
medium-diverse-v2	0.0	8.0	0.2	0.0±0.0	0.2±0.4	20.6±27.7	46.6±24.0	72.8±7.0	**79.1**±5.6
large-play-v2	6.7	0.0	0.0	0.0±0.0	0.0±0.0	2.2±3.8	16.4±17.1	**42.0**±3.8	35.3±8.3
large-diverse-v2	2.2	0.0	0.0	0.0±0.0	0.0±0.0	3.0±6.7	3.2±4.1	**46.0**±4.5	36.3±13.7
AntMaze total	142.8	142.0	91.6	95.4±15.5	120.2±41.7	164.6±84.3	218.0±59.2	**392.4**±30.5	359.6±39.7

State-of-the-art method

Analysis on Support Constraint



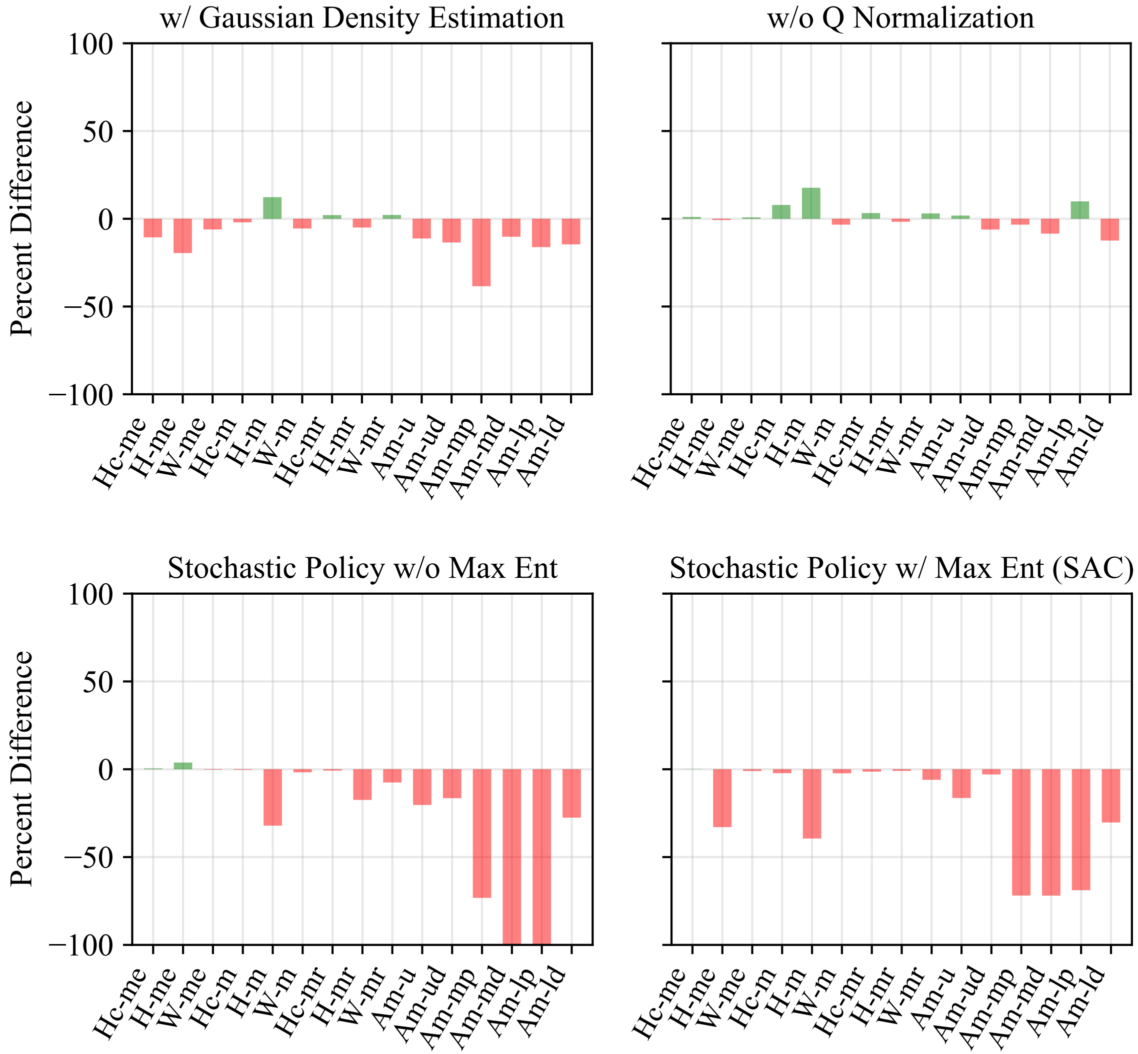
(a) Effect of λ.

(b) Tradeoff between constraint strength and optimality.

- Regularization weight λ effectively applies support constraint with different strength.
- With varying levels of constraint strength, SPOT always achieve the strongest performance among extensive policy constraint methods.

Ablation Study

- Gaussian density estimation degrades the performance on datasets with complex behaviors.
- Q normalization makes an insignificant impact on total performance
- TD3 may be preferable with native designs addressing function approximation error



Online Fine-tuning on D4RL-AntMaze

- Strong offline2online performance
- A minimal gap between offline RL and well-established online RL methods

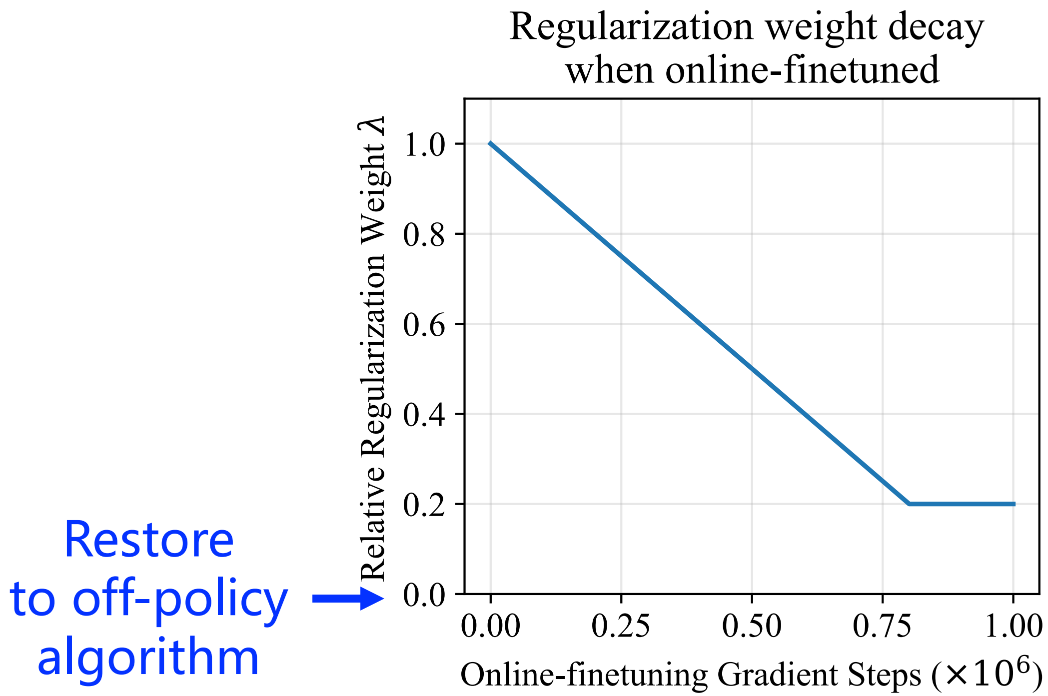


Table 4: Online fine-tuning results on AntMaze tasks, showing initial performance after offline RL and performance after 1M steps of online RL. All numbers are reported by the mean of 5 seeds.

	IQL	SPOT (Ours)
umaze-v2	85.4 → 96.2	**93.2** → **99.2** (+3.0)
umaze-diverse-v2	**70.8** → 62.2	41.6 → **96.0** (+33.8)
medium-play-v2	68.6 → 89.8	**75.2** → **97.4** (+7.6)
medium-diverse-v2	**73.4** → 90.2	73.0 → **96.2** (+6.0)
large-play-v2	40.0 → 78.6	**40.8** → **89.4** (+10.8)
large-diverse-v2	40.4 → 73.4	**44.0** → **90.8** (+17.4)
AntMaze total	**378.6** → 490.4	367.8 → **569.0** (+78.6)

Computational Efficiency

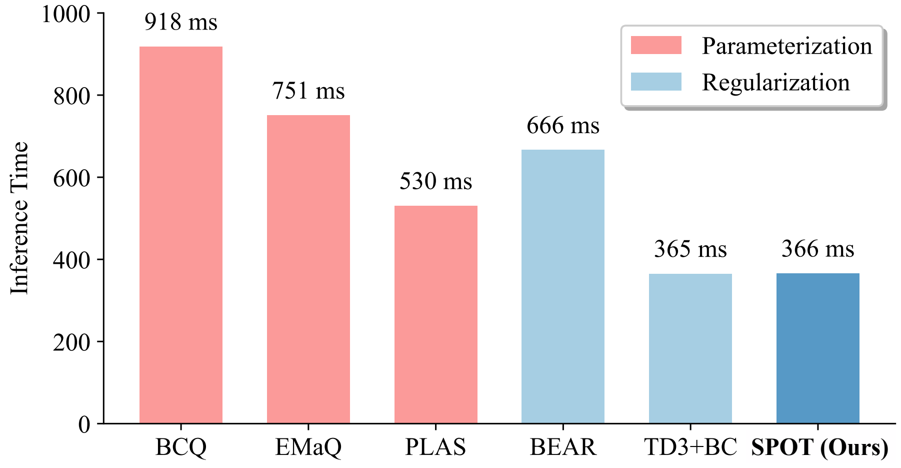


Figure 3: Runtime of various offline RL algorithms interacting with the HalfCheetah environment to produce a 1000-steps trajectory. See Appendix C.5 for the details.

Table 9: Train time of 1M steps of various offline RL algorithms.

	BCQ	BEAR	PLAS	CQL	TD3+BC	SPOT
Train time	5h 25m	12h 30m	3h 5m	14h 20m	1h 58m	3h 25m

- One forward pass of the policy network to do inference
- Indeed add training overhead due to the VAE-based density estimator

Summary

- **Benefits**
  - Excellent offline RL performance
  - Strong offline2online RL performance
  - Computational efficiency at inference

① Pluggable regularization

② Explicit Behavior Estimation

- **Limitations**
  - Limited by the accuracy of behavior policy estimation
    - Future work: stronger generative model
  - Hyperparameter selection with online evaluation
    - Future work: offline policy evaluation, offline manual or auto-tuning

Thank You!

wujialong0229@gmail.com