香港中文大學
The Chinese University of Hong Kong

ICLR

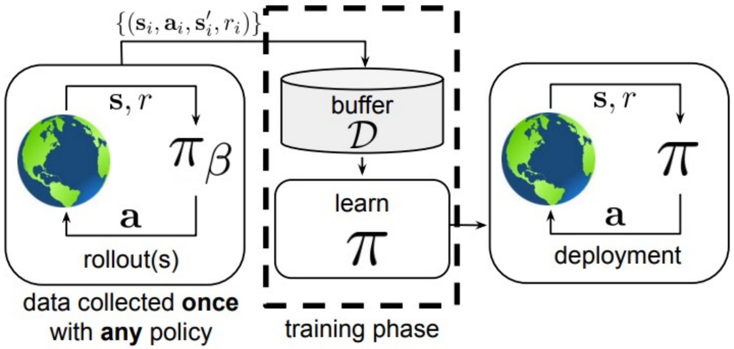
DMBP: Diffusion Model-Based Predictor for Robust Offline Reinforcement Learning against State Observation Perturbations

Zhihe Yang, Yunjian Xu*

The Chinese University of Hong Kong
*Corresponding author

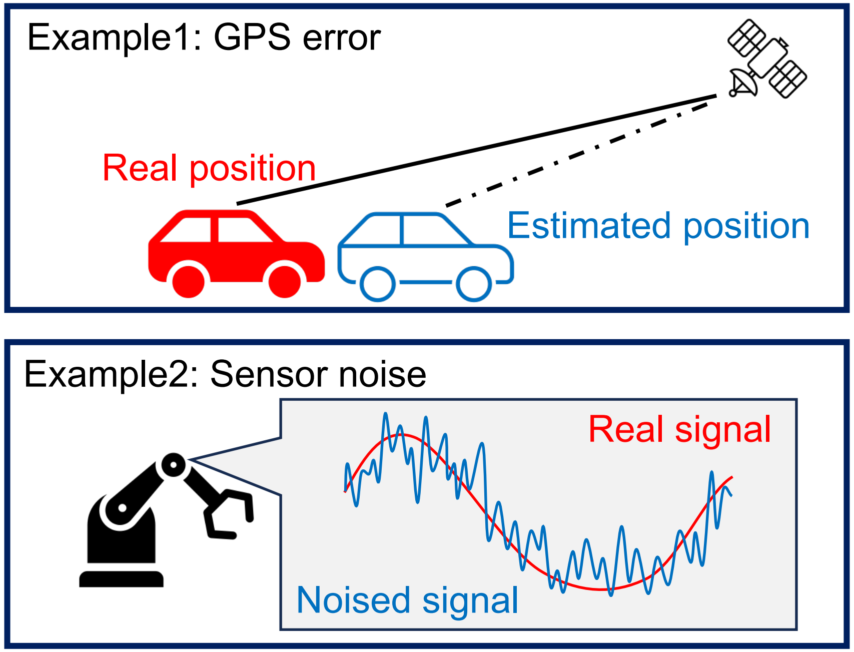
Offline Reinforcement Learning

Diagram of Offline RL[1] :



One challenge for real-world application of offline RL:

State observation perturbations

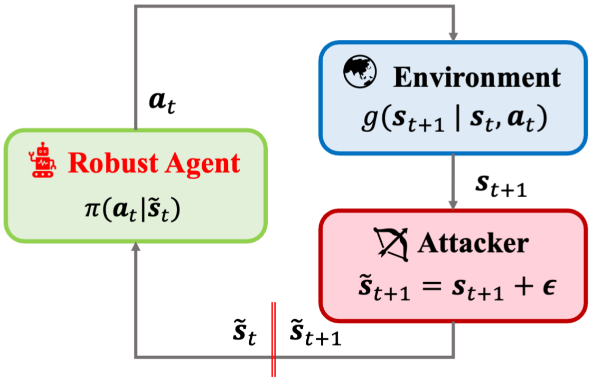


[1] Levine, Sergey, et al. "Offline reinforcement learning: Tutorial, review, and perspectives on open problems." *arXiv preprint arXiv:2005.01643* (2020).

State observation perturbations

Classical solution:

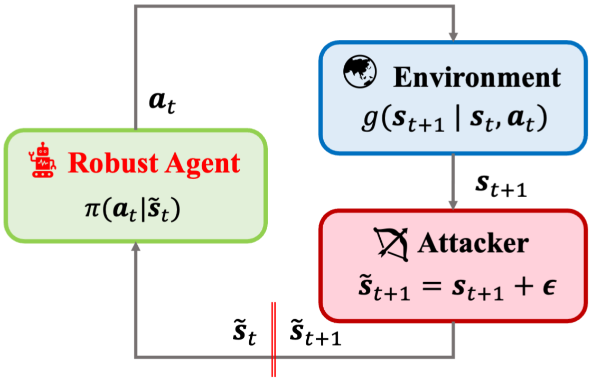
- **Smooth policy**
  overconservative and sensitive to noise scales
- **Adversarially trained policy:**
  not applicable in offline training manner



State observation perturbations

Classical solution:

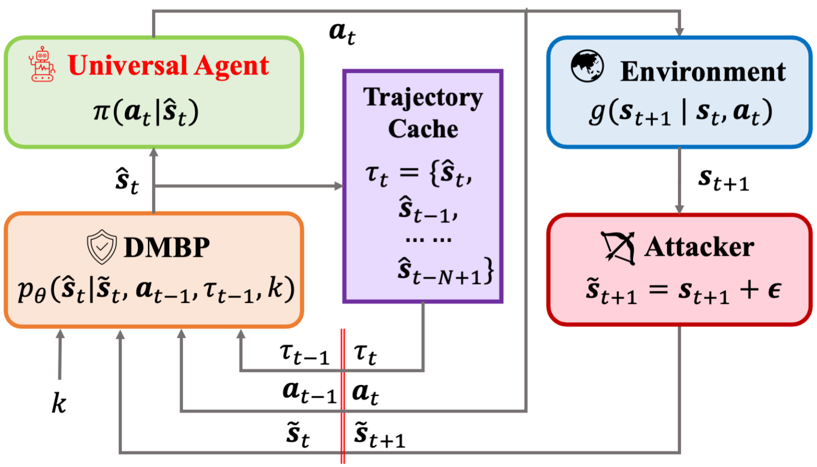
- **Smooth policy**
  overconservative and sensitive to noise scales
- **Adversarially trained policy:**
  not applicable in offline training manner



Our solution (**Diffusion Model-Based Predictor**):
Recover the actual states for decision-making.

Main challenge: Error accumulation

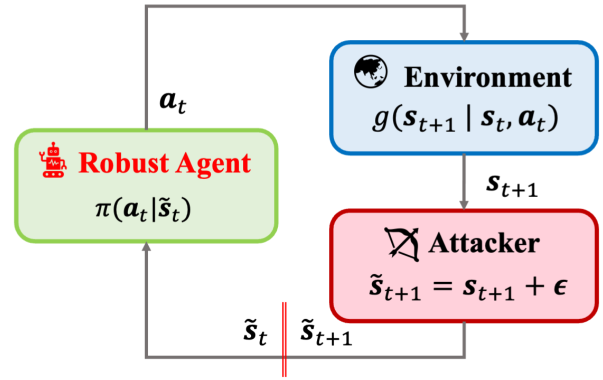
Our contribution:
- Novel framework for robust RL
- Non-Markovian loss function



State observation perturbations

Classical solution:

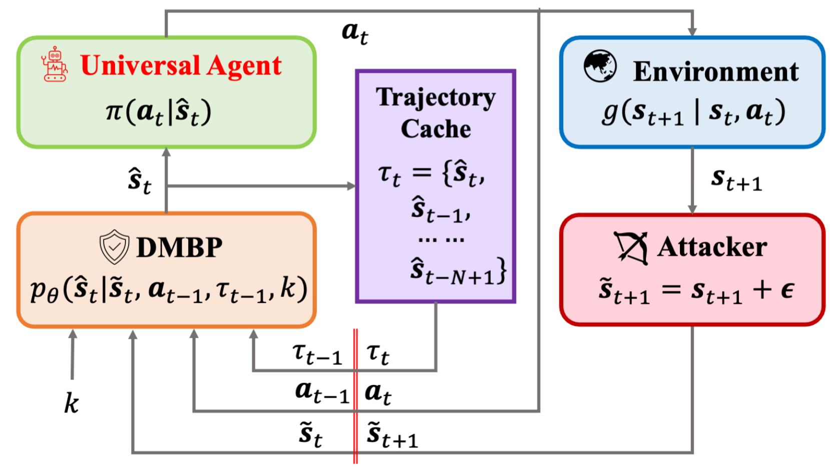
- **Smooth policy**
  overconservative and sensitive to noise scales
- **Adversarially trained policy:**
  not applicable in offline training manner



Our solution (Diffusion Model-Based Predictor):
Recover the actual states for decision-making.

Main challenge: Error accumulation

Our contribution:
- Novel framework for robust RL
- Non-Markovian loss function



Advantages of DMBP:
- Combinable with any offline RL algorithms.
- Applicable for different scales of noises.
- Applicable for incomplete state observations with unobserved dimensions.
- Does not lead to over-conservative policy.

DMBP for Predicting Real States

Notation: $\boldsymbol{s}_t$ — original state; $\tilde{\boldsymbol{s}}_t$ — noised state; $\hat{\boldsymbol{s}}_t$ — recovered state.
Superscript i — diffusion timesteps; Subscript t — RL timesteps.

Sampling of denoised state $\hat{\boldsymbol{s}}_t$:

$$\begin{aligned}\hat{\boldsymbol{s}}_t &\sim p_\theta(\tilde{\boldsymbol{s}}_t^{0:k} \mid \boldsymbol{a}_{t-1}, \boldsymbol{\tau}_{t-1}^{\hat{\boldsymbol{s}}}) \\ &= f_k(\tilde{\boldsymbol{s}}_t) \prod_{i=1}^{k} p_\theta(\tilde{\boldsymbol{s}}_t^{i-1} \mid \tilde{\boldsymbol{s}}_t^{i}, \boldsymbol{a}_{t-1}, \boldsymbol{\tau}_{t-1}^{\hat{\boldsymbol{s}}})\end{aligned}$$

where $f_k(\tilde{\boldsymbol{s}}_t) = \sqrt{\bar{\alpha}_k}\tilde{\boldsymbol{s}}_t$.

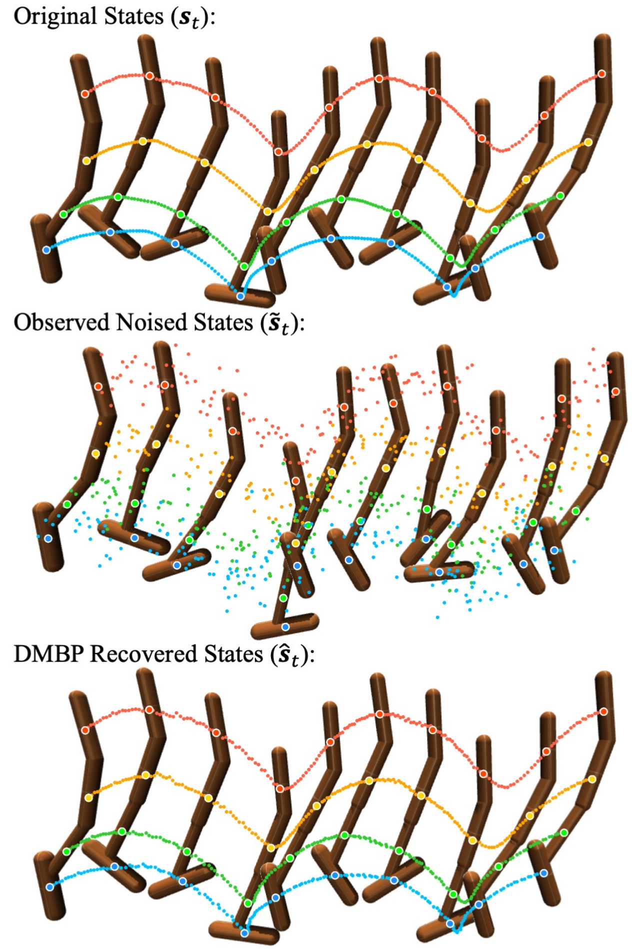
Reverse diffusion chain:

Following [1], we model transitions $p_\theta(\tilde{\boldsymbol{s}}_t^{i-1} \mid \tilde{\boldsymbol{s}}_t^{i}, \boldsymbol{a}_{t-1}, \boldsymbol{\tau}_{t-1}^{\hat{\boldsymbol{s}}})$ as Gaussian process:

$$\begin{aligned}&\tilde{\boldsymbol{s}}_t^{i-1} \mid \tilde{\boldsymbol{s}}_t^{i} \\ &= \frac{\tilde{\boldsymbol{s}}_t^{i}}{\sqrt{\alpha_i}} - \frac{\beta_i}{\sqrt{\alpha_i(1-\bar{\alpha}_i)}}\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{s}}_t^{i}, \boldsymbol{a}_{t-1}, \boldsymbol{\tau}_{t-1}^{\hat{\boldsymbol{s}}}, i) + \sqrt{\tilde{\beta}_i}\boldsymbol{\epsilon}\end{aligned}$$

where $\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{s}}_t^{i}, \boldsymbol{a}_{t-1}, \boldsymbol{\tau}_{t-1}^{\hat{\boldsymbol{s}}}, i)$ is the neuron-network predicted noise.

hopper-medium-replay-v2
(Gaussian Noise with std of 0.10)



[1] Ho, Jonathan, et, al. "Denoising diffusion probabilistic models." *Advances in neural information processing systems* 33 (2020): 6840-6851.

Non-Markovian Loss (Main Contribution)

Problem with classical training of diffusion models:

The accuracy of the current denoising result $\hat{\boldsymbol{s}}_t$ is highly dependent on the accuracy of the diffusion condition $\boldsymbol{\tau}_{t-1}^{\hat{\boldsymbol{s}}}$.

$$\left.\begin{cases} \text{Training phase:} & \boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{s}}_t^i, \boldsymbol{a}_{t-1}, \boldsymbol{\tau}_{t-1}^{\boldsymbol{s}}, i) \\ \text{Testing phase:} & \boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{s}}_{t+1}^i, \boldsymbol{a}_t, \boldsymbol{\tau}_{t-1}^{\hat{\boldsymbol{s}}}, i) \end{cases}\right\}$$

Severe error-accumulation

Our Proposed Non-Markovian training objective:

$$\mathcal{L}_{\text{entropy}} = \sum_{t=2}^{T} \mathbb{E}_{\boldsymbol{s}_t \in \boldsymbol{\tau}, q(\boldsymbol{s}_t)} \left[-\log P(\hat{\boldsymbol{s}}_t \mid \boldsymbol{a}_{t-1}, \boldsymbol{\tau}_{t-1}^{\hat{\boldsymbol{s}}}) \right]$$

Along RL Trajectory — Condition on denoised trajectory

Closed form expression:

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{\boldsymbol{s}_1 \sim d_0, \boldsymbol{\epsilon}_t^i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), i \sim \mathcal{U}_K} \left[\sum_{t=2}^{T} \| \boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{s}}_t^i, \boldsymbol{a}_{t-1}, \boldsymbol{\tau}_{t-1}^{\hat{\boldsymbol{s}}}, i) - \boldsymbol{\epsilon}_t^i \|^2 \right],$$

$$\boldsymbol{\tau}_{t-1}^{\hat{\boldsymbol{s}}} := \{ \hat{\boldsymbol{s}}_j \mid j \le t-1 \}$$

$$\hat{\boldsymbol{s}}_j = \begin{cases} \boldsymbol{s}_1 & \text{if } j = 1, \\ f_k(\tilde{\boldsymbol{s}}_j^k) \prod_{i=1}^{k} p_\theta(\tilde{\boldsymbol{s}}_j^{i-1} \mid \tilde{\boldsymbol{s}}_j^i, \boldsymbol{a}_{j-1}, \boldsymbol{\tau}_{j-1}^{\hat{\boldsymbol{s}}}) & \text{otherwise } (j \in \{2, \ldots, t-1\}). \end{cases}$$

Non-Markovian Loss (Main Contribution)

Practical Loss function:

(N : condition trajectory length M : sample trajectory length)

$$\mathcal{L}(\theta) = \mathbb{E}_{i\sim\mathcal{U}_K, \boldsymbol{\epsilon}_t\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}),(\boldsymbol{s}_{t-N},\dots,\boldsymbol{s}_{t+M-1})\in\mathcal{D}_\nu}\Bigg[\underbrace{\|\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{s}}_t^i, \boldsymbol{a}_{t-1}, \boldsymbol{\tau}_{t-1}^{\boldsymbol{s}}, i) - \boldsymbol{\epsilon}_t^i\|^2}_{L_t} + \sum_{m=t+1}^{t+M-1}\underbrace{\|\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{s}}_m^i, \boldsymbol{a}_{m-1}, \boldsymbol{\tau}_{m-1}^{\breve{\boldsymbol{s}}}, i) - \boldsymbol{\epsilon}_m^i\|^2}_{L_m}\Bigg],$$

Classical loss of diffusion models → L_t

Additional term of our non-Markovian loss → $\sum_{m=t+1}^{t+M-1} L_m$

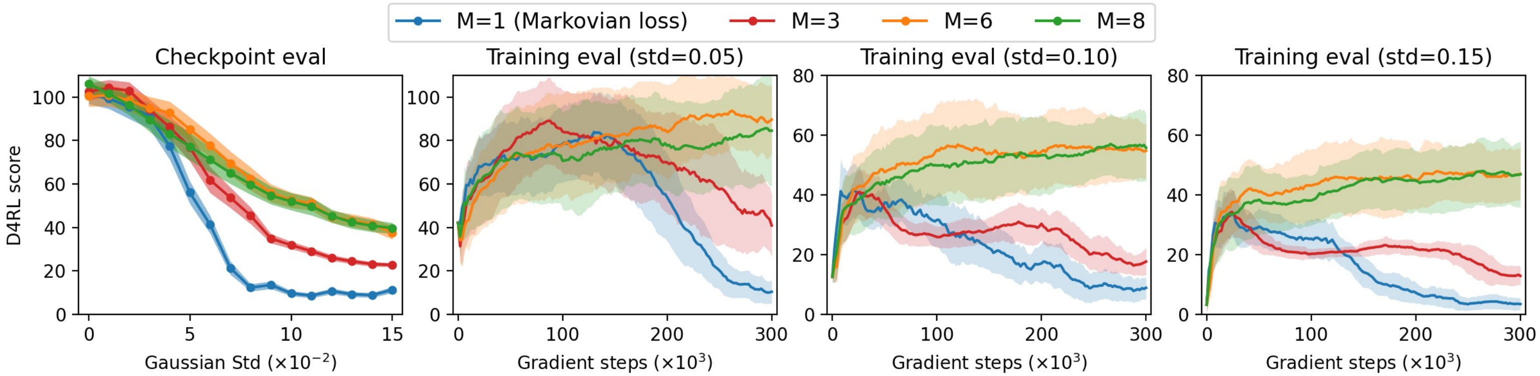
Trajectory condition for predictor $\boldsymbol{\epsilon}_\theta$:

- in L_t: $\boldsymbol{\tau}_{t-1}^{\boldsymbol{s}} = \{\boldsymbol{s}_{t-N}, \dots, \boldsymbol{s}_{t-1}\}$
- in L_m: $\boldsymbol{\tau}_{m-1}^{\breve{\boldsymbol{s}}} = \{\breve{\boldsymbol{s}}_j \mid j \in \{m-N, \dots, m-1\}\}$

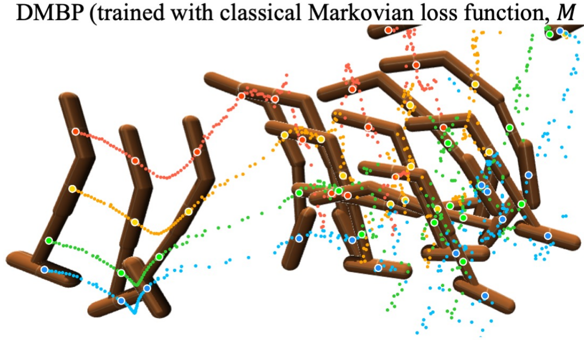
$$\breve{\boldsymbol{s}}_j = \begin{cases} \boldsymbol{s}_j & \text{if } j < t, \\ \frac{1}{\sqrt{\bar{\alpha}_i}}\left[\tilde{\boldsymbol{s}}_j^i - \sqrt{1-\bar{\alpha}_i}\,\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{s}}_j^i, \boldsymbol{a}_{j-1}, \boldsymbol{\tau}_{j-1}^{\breve{\boldsymbol{s}}}, i)\right] & \text{otherwise } (j \in \{t, \dots, t+M-2\}). \end{cases}$$

Ablation Study

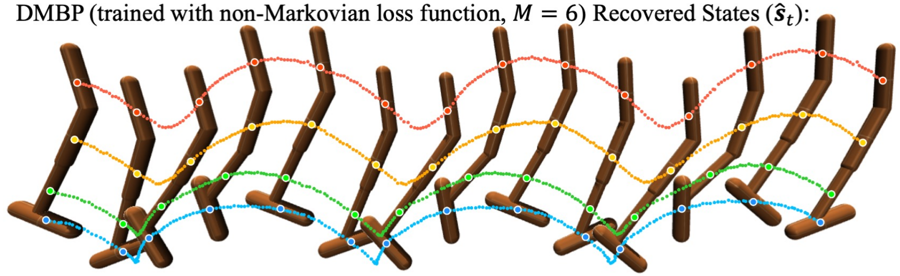
hopper-expert-v2 (prone to error accumulation)



DMBP (trained with classical Markovian loss function, $M = 1$) Recovered States ($\hat{\boldsymbol{s}}_t$):

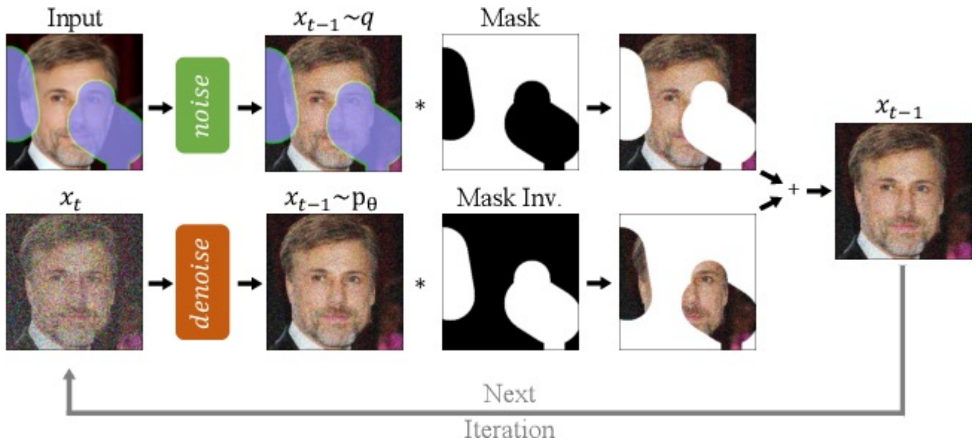


DMBP (trained with non-Markovian loss function, $M = 6$) Recovered States ($\hat{\boldsymbol{s}}_t$):



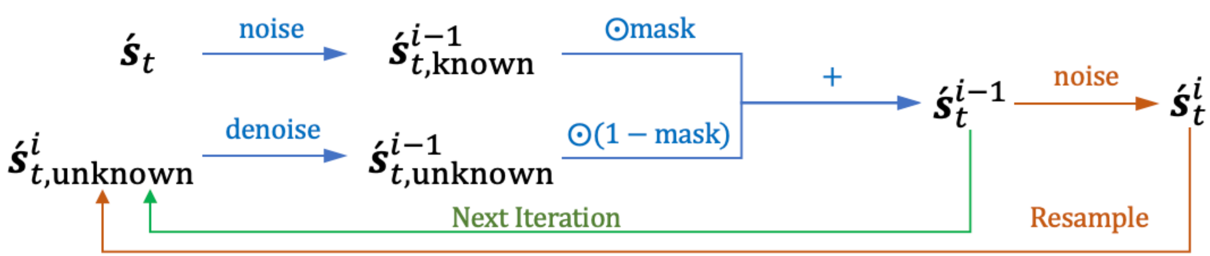
State Infilling for Unobserved Dimensions

Diffusion models have been proved to be successful in **Image inpainting** tasks[1]:



Real world circumstance: compromised sensors ($\acute{s}_t = s_t \odot \mathbf{mask}$)

Recover the missing part of state observation utilizing "resample" technique[1].



[1] Lugmayr, Andreas, et al. "Repaint: Inpainting using denoising diffusion probabilistic models." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.

Experiment（Noised Observations）

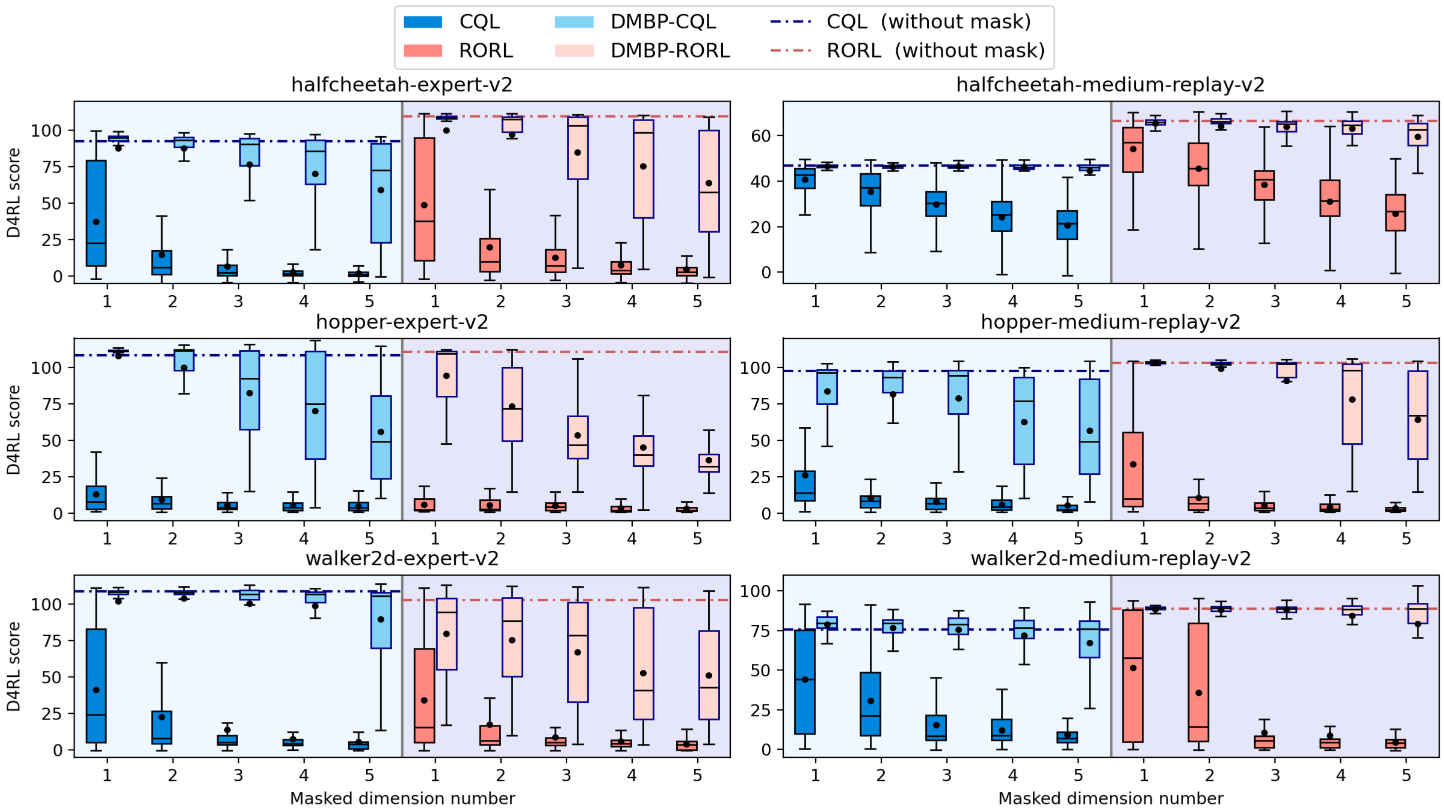
➢ Gaussian random noises

Env	Dataset	Noise scale	BCQ base	BCQ DMBP	CQL base	CQL DMBP	TD3+BC base	TD3+BC DMBP	Diffusion QL base	Diffusion QL DMBP	RORL base	RORL DMBP
HalfCheetah	e	0	96.9±1.8	-	93.0±6.1	-	95.8±8.9	-	92.9±10.7	-	108.5±11.2	-
		0.05	4.5±2.6	60.2±23.9	18.1±8.6	60.9±22.5	7.3±6.6	77.1±15.5	4.8±3.6	75.2±20.7	15.4±3.9	55.7±29.2
		0.10	4.5±2.5	26.8±16.2	7.4±4.0	40.5±16.6	4.7±3.6	47.5±22.2	3.3±2.5	39.8±21.8	3.7±1.9	32.8±20.4
	m-r	0	41.6±4.2	-	47.0±1.0	-	45.2±0.9	-	47.7±0.8	-	66.7±1.4	-
		0.10	20.6±6.9	38.5±11.2	35.6±1.3	45.8±1.0	28.5±5.5	44.3±1.0	30.1±4.1	45.6±0.9	43.5±2.4	61.9±1.2
		0.15	14.8±10.2	35.1±8.7	28.8±1.5	44.6±1.1	24.0±8.9	42.5±2.6	24.2±7.6	44.6±3.0	30.3±5.9	58.4±1.2
Hopper	e	0	88.4±22.4	-	109.1±13.7	-	108.9±10.5	-	104.9±15.1	-	110.4±3.1	-
		0.05	34.3±13.4	61.0±25.2	41.2±21.8	85.7±26.2	32.2±18.4	79.1±28.2	38.2±12.4	84.8±27.4	56.9±34.9	64.3±19.8
		0.10	24.3±10.9	37.1±18.5	24.3±11.8	48.8±20.4	22.7±11.6	32.6±18.7	24.0±9.3	56.1±17.3	24.1±20.2	37.5±10.5
	m-r	0	78.7±19.6	-	96.9±8.8	-	80.9±24.5	-	95.7±17.2	-	103.1±0.8	-
		0.10	15.7±9.0	66.8±17.3	47.5±21.6	89.1±12.4	14.4±12.3	71.9±24.5	25.9±12.4	85.9±20.9	85.9±29.5	103.2±1.3
		0.15	11.1±7.2	64.5±17.2	33.7±21.2	80.7±16.5	9.6±7.3	66.1±22.8	17.9±11.5	72.2±22.9	51.1±22.3	104.2±3.2
Walker2d	e	0	111.6±0.6	-	108.8±1.9	-	110.7±0.5	-	109.6±0.5	-	104.8±12.5	-
		0.10	77.9±37.6	110.3±2.0	97.6±21.9	94.3±20.3	72.9±39.4	109.2±1.5	93.3±27.2	109.1±4.0	95.4±19.7	97.8±20.2
		0.15	28.2±32.4	104.2±13.5	78.9±33.2	83.4±23.3	9.2±13.6	107.5±5.2	30.5±32.5	94.5±18.1	81.6±26.4	84.5±26.4
	m-r	0	50.6±31.6	-	79.9±4.8	-	84.7±9.8	-	93.1±10.9	-	88.7±1.9	-
		0.10	14.7±11.1	53.1±28.5	70.8±18.9	78.7±7.2	40.7±25.3	84.4±8.7	59.6±31.8	92.6±10.6	88.6±1.1	88.4±2.5
		0.15	11.2±5.9	52.9±29.9	48.6±26.5	73.6±10.1	16.5±12.8	77.9±17.2	19.2±15.7	91.3±9.6	89.4±1.2	89.0±4.5

➢ Uniform random noises (U-rand), Maximum action difference attack (MAD), and Minimum Q-value attack (MinQ)

Env	Dataset/ Noise Scale	Noise Type	BCQ base	BCQ DMBP	CQL base	CQL DMBP	TD3+BC base	TD3+BC DMBP	Diffusion QL base	Diffusion QL DMBP	RORL base	RORL DMBP
HalfCheetah	e 0.05	U-rand	7.4±4.9	69.1±21.5	27.2±6.4	69.6±22.4	16.3±13.1	84.2±17.1	11.6±10.9	77.8±21.8	24.3±7.5	66.8±27.0
		MAD	3.6±1.7	52.5±17.9	12.4±6.9	61.2±19.7	4.7±3.5	65.4±16.0	4.3±3.2	62.9±13.2	14.1±2.5	54.3±27.1
		MinQ	12.8±9.3	51.8±23.9	19.4±11.3	60.4±19.4	18.0±4.2	88.2±11.3	8.0±6.7	71.1±15.2	9.3±8.8	71.0±29.1
	m-r 0.10	U-rand	31.5±10.6	40.3±5.9	40.9±2.6	46.4±1.8	36.9±6.6	46.9±1.1	38.5±5.7	46.8±0.9	39.9±2.3	61.2±1.1
		MAD	19.2±8.2	29.4±6.9	29.0±2.6	46.5±0.9	27.1±3.4	36.2±0.9	22.3±3.8	34.5±5.5	22.5±1.5	62.3±1.0
		MinQ	5.1±5.2	36.7±8.8	39.2±0.8	46.2±1.1	36.7±6.8	44.8±1.1	37.0±4.8	38.6±1.1	34.0±1.4	63.2±2.3
Hopper	e 0.05	U-rand	46.1±20.7	66.9±26.3	59.6±29.4	95.7±23.8	42.6±28.4	84.0±27.4	53.2±20.8	84.4±25.3	85.3±37.0	81.9±25.2
		MAD	31.1±14.4	53.2±24.2	22.6±13.9	73.9±27.9	27.2±10.9	60.3±27.2	36.8±9.0	37.1±12.3	36.6±22.2	59.0±13.8
		MinQ	47.4±18.9	62.5±27.9	32.7±13.5	58.7±17.9	45.3±27.5	95.7±27.6	66.7±33.6	59.2±23.9	79.8±32.7	59.4±22.1
	m-r 0.10	U-rand	18.5±8.2	68.9±19.2	66.3±20.1	95.9±8.8	20.6±9.1	65.4±22.0	33.9±10.7	94.9±17.7	80.7±28.0	103.5±1.5
		MAD	5.1±5.0	37.5±26.1	32.1±15.9	88.9±13.7	6.1±5.5	64.3±21.8	9.9±8.1	38.3±15.8	51.6±30.7	97.5±2.5
		MinQ	5.3±5.4	18.3±18.4	84.6±14.1	87.5±6.6	11.8±7.6	80.5±18.1	51.2±25.1	62.5±27.3	98.3±6.2	103.2±2.4
Walker2d	e 0.10	U-rand	102.1±1.8	110.4±0.8	106.1±9.9	106.0±7.4	106.1±2.9	110.0±0.5	107.2±1.0	109.4±0.5	95.1±15.7	97.2±9.5
		MAD	50.5±43.7	70.5±13.3	64.1±27.0	97.6±16.1	19.9±22.7	69.7±17.5	36.6±35.5	88.2±24.8	61.9±29.2	83.8±19.9
		MinQ	99.9±22.2	105.6±1.1	99.9±11.8	102.4±6.9	91.9±22.4	105.5±1.3	101.1±2.0	102.4±1.3	91.8±28.0	89.3±13.3
	m-r 0.15	U-rand	17.3±12.2	54.9±25.7	69.2±20.9	78.1±9.2	51.2±28.3	83.6±14.8	64.2±27.8	91.1±12.1	89.9±1.1	88.7±2.1
		MAD	6.6±3.3	43.4±29.8	19.7±14.7	78.4±8.8	8.8±4.4	70.8±19.1	7.2±2.3	66.1±24.2	81.9±11.5	90.5±3.5
		MinQ	7.3±4.2	30.3±26.1	66.5±11.8	78.5±4.2	21.7±15.9	76.4±14.9	47.2±23.2	68.0±19.5	82.3±1.4	89.6±1.7

Experiment（Incomplete Observations）



Baseline Algorithm:

[1] Fujimoto, Scott, et al. "Off-policy deep reinforcement learning without exploration." *International conference on machine learning*. PMLR, 2019.
[2] Kumar, Aviral, et al. "Conservative q-learning for offline reinforcement learning." *Advances in Neural Information Processing Systems* 33 (2020): 1179-1191.
[3] Fujimoto, Scott, et al. "A minimalist approach to offline reinforcement learning." *Advances in neural information processing systems* 34 (2021): 20132-20145.
[4] Wang, Zhendong, et al. "Diffusion policies as an expressive policy class for offline reinforcement learning." *arXiv preprint arXiv:2208.06193* (2022).
[5] Yang, Rui, et al. "Rorl: Robust offline reinforcement learning via conservative smoothing." *Advances in neural information processing systems* 35 (2022): 23851-23866.

Thanks for your listening!

paper

code