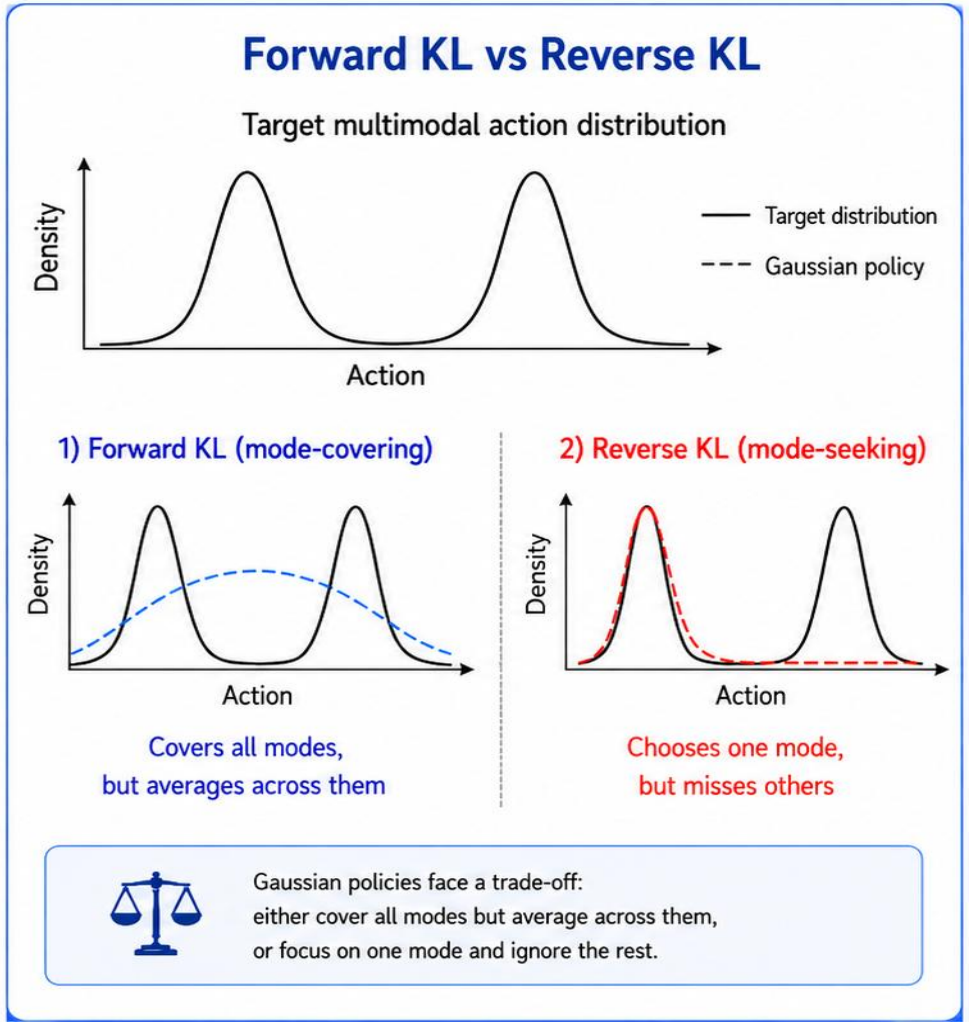
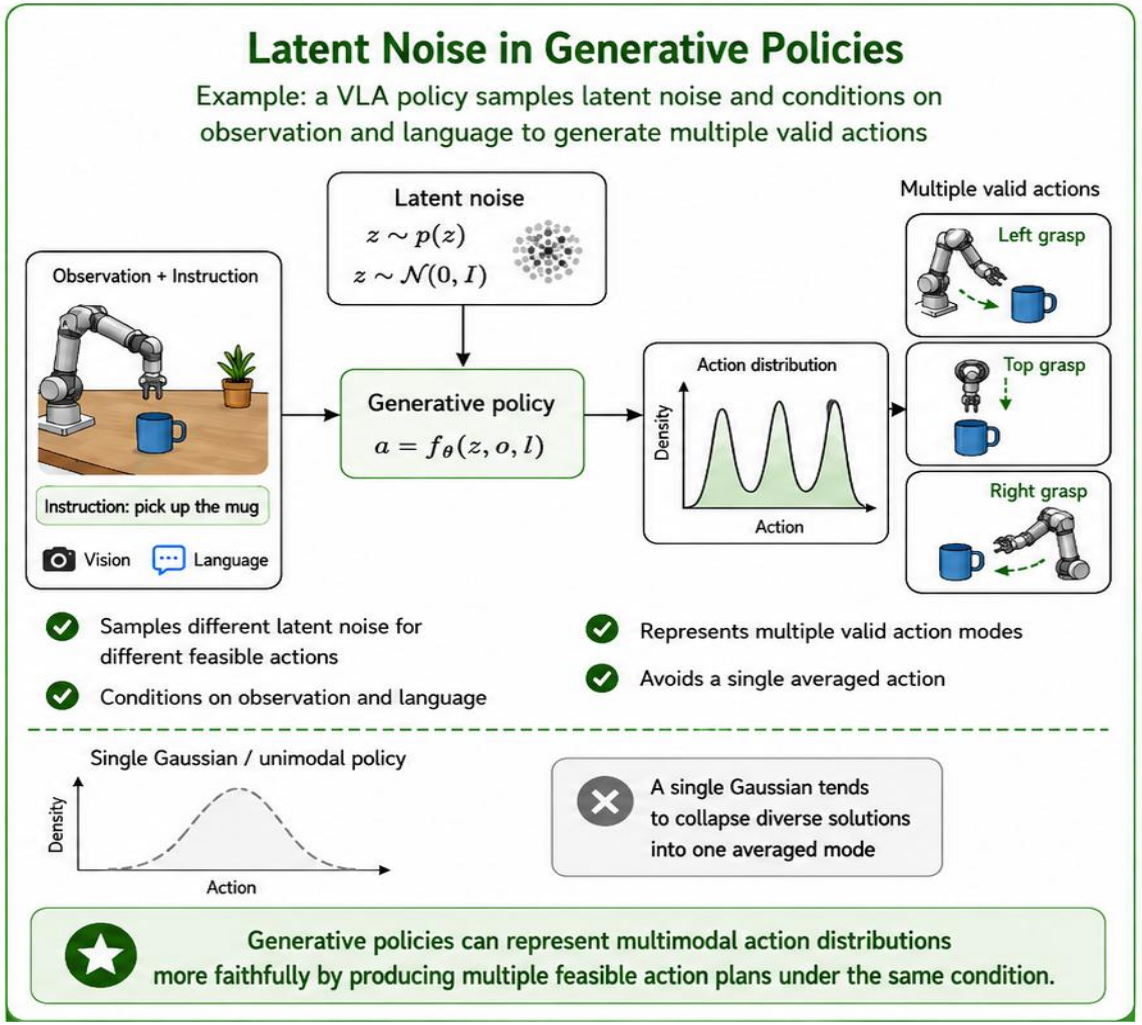
Generative Models For RL

Background





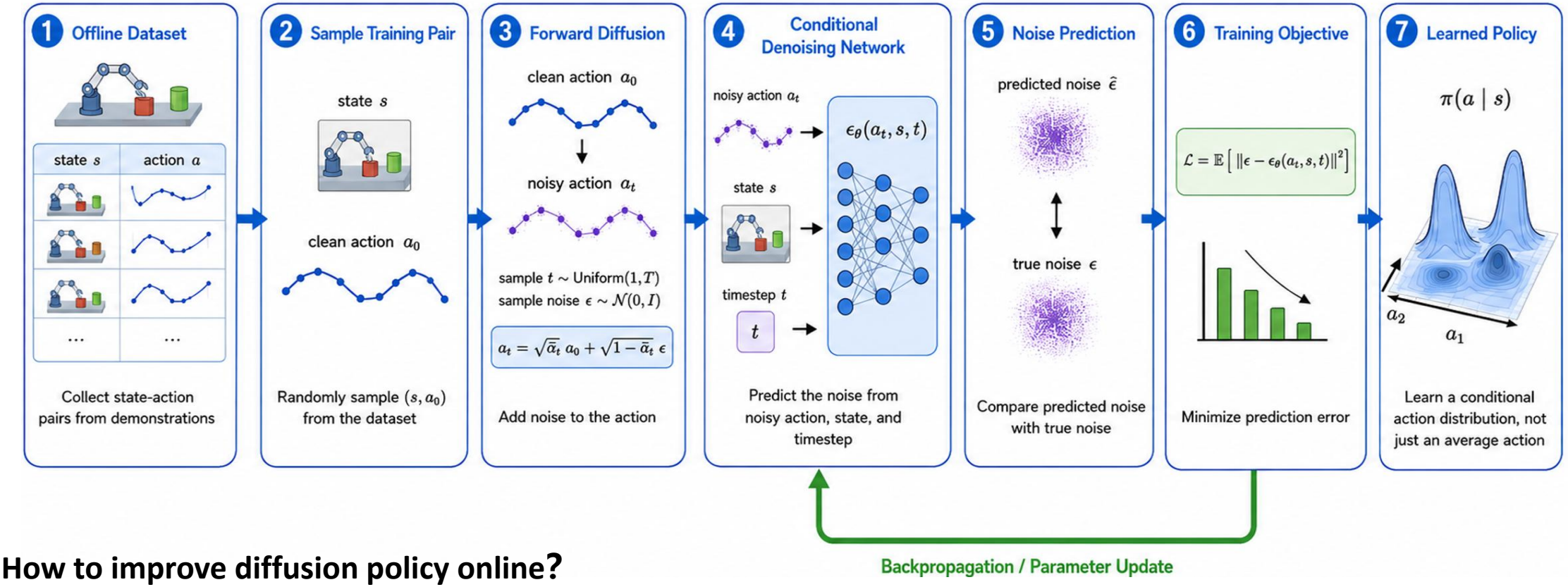
Generative policies better capture *multimodal action distributions*.

Steering Your Diffusion Policy with Latent Space Reinforcement Learning

Andrew Wagenmaker* UC Berkeley
Mitsuhiko Nakamoto* UC Berkeley
Yunchu Zhang* University of Washington
Seohong Park UC Berkeley
Waleed Yagoub University of Washington
Anusha Nagabandi Amazon
Abhishek Gupta* University of Washington
Sergey Levine* UC Berkeley

Motivation

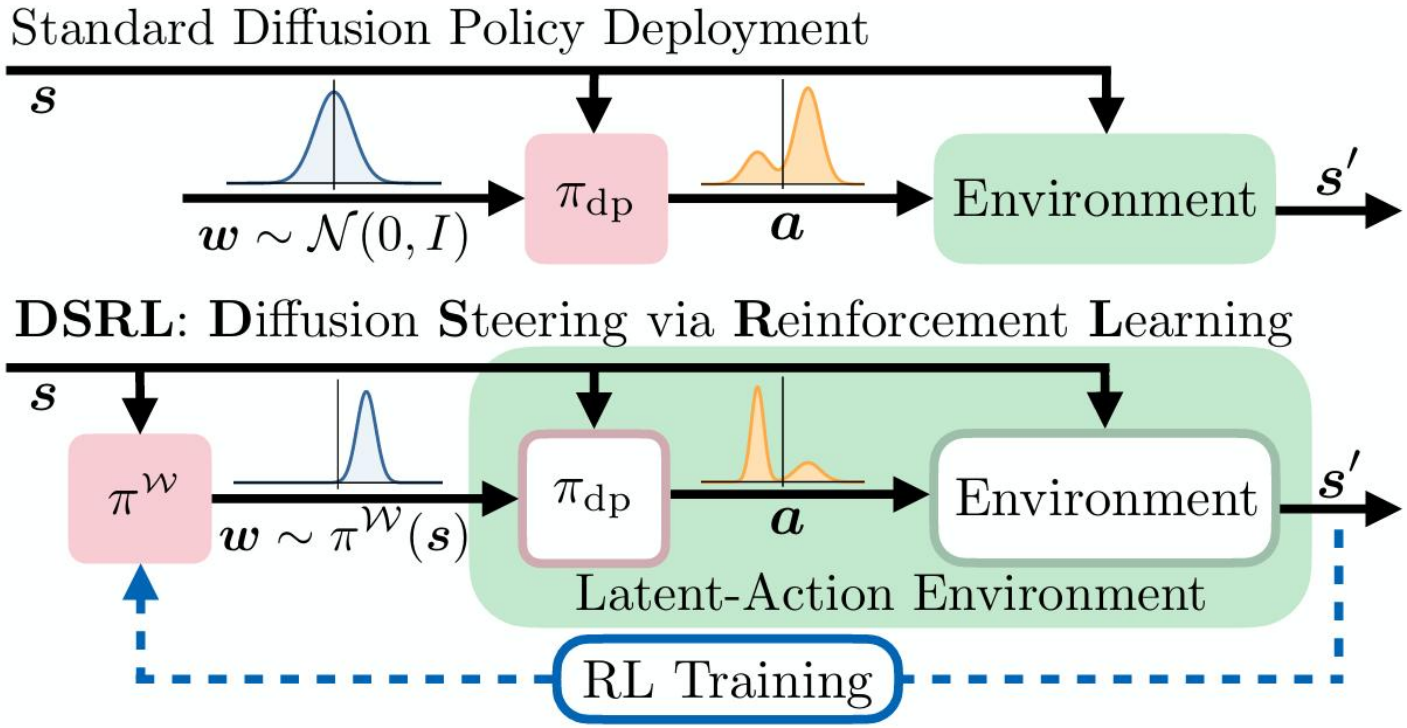
Training process of diffusion policy

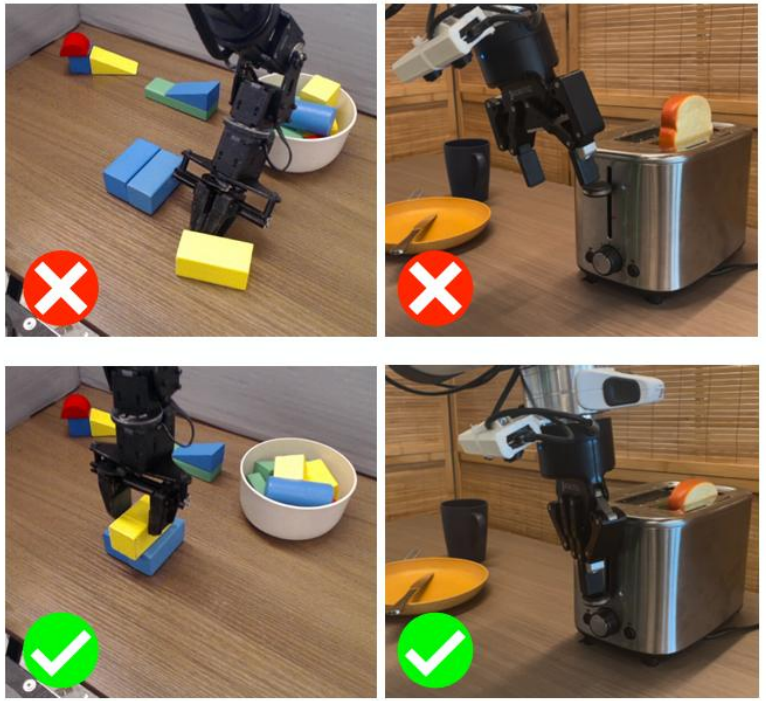


How to improve diffusion policy online?

- Collect more expert demonstrations ⟶ costly and time-consuming
- Online RL ⟶ computationally heavy, unstable, and may destroy the pretrained behavior prior

Method





$$a \sim \pi_{\text{dp}}^{\mathcal{W}}(s, w) \qquad w \sim \mathcal{N}(0, I) \qquad w_t \sim \pi^{\mathcal{W}}(s_t)$$

Optimize the pre-trained policy by noise manipulation.

Method

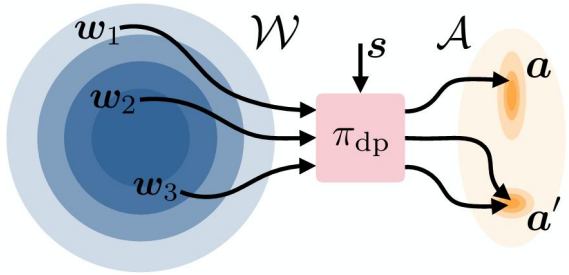
DSRL-SAC

$$w \sim \pi_\phi^{\mathcal{W}}(w|s)$$

$$a = \pi_{\text{dp}}^{\omega}(s, w)$$

$$(s, w, r, s')$$

$$Q^{\mathcal{W}}(s, w)$$



DSRL-NA

$$w' \neq w, \qquad \pi_{\text{dp}}^{\omega}(s, w') \approx \pi_{\text{dp}}^{\omega}(s, w)$$

$$Q^{\mathcal{A}}(s, a) \longrightarrow Q^{\mathcal{W}}(s, w)$$

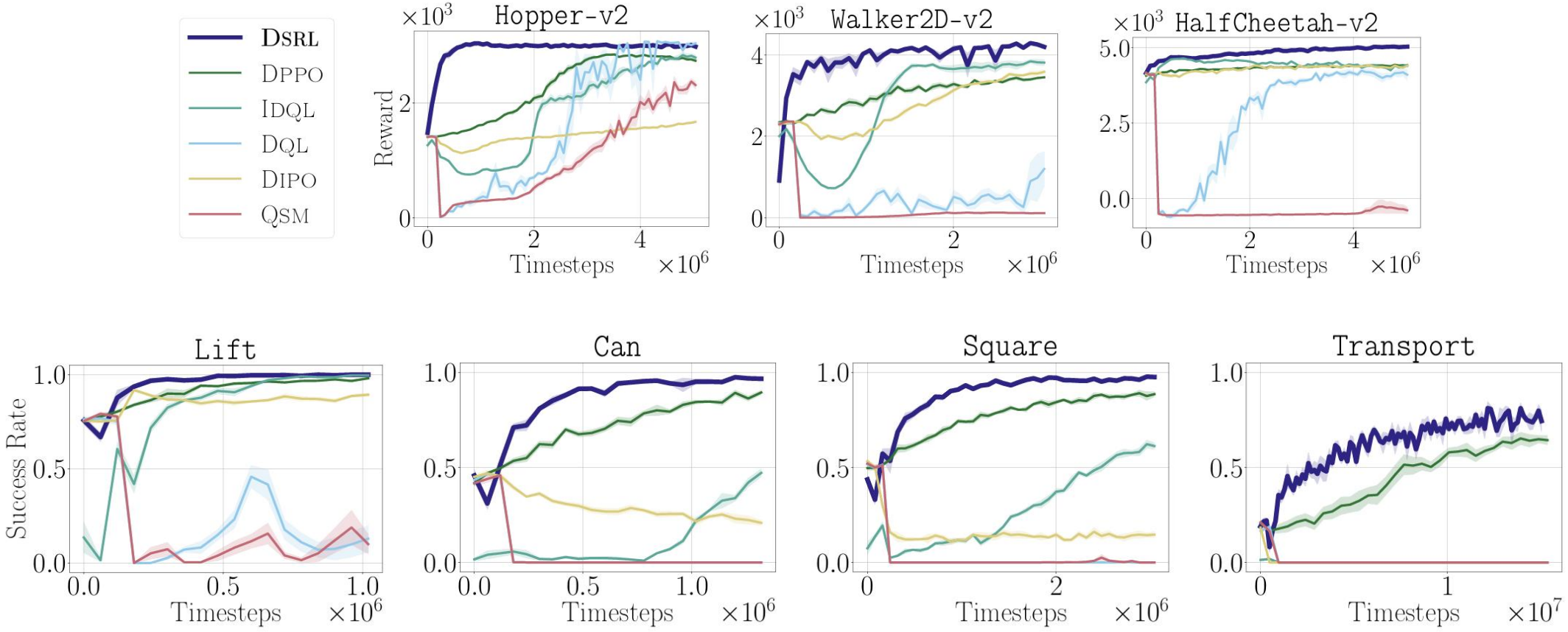
Critic distillation

Algorithm 1 Noise-Aliased Diffusion Steering via Reinforcement Learning (DSRL-NA)

input: pretrained diffusion policy $\pi_{\text{dp}}^{\mathcal{W}}$, offline data $\mathcal{D}_{\text{off}}$ and/or online environment $\mathcal{M}$
Initialize replay buffer $\mathcal{B} \leftarrow \mathcal{D}_{\text{off}}$, $\mathcal{A}$-critic $Q^{\mathcal{A}}$, latent-noise critic $Q^{\mathcal{W}}$, latent-noise actor $\pi^{\mathcal{W}}$
for $t = 1, \ldots, T$ **do**
  Update $Q^{\mathcal{A}}$: $\min_{Q^{\mathcal{A}}} \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a},r,\boldsymbol{s}')\sim\mathcal{B},\boldsymbol{a}'\sim\pi_{\text{dp}}^{\mathcal{W}}(\boldsymbol{s}',\pi^{\mathcal{W}}(\boldsymbol{s}'))}\left[(Q^{\mathcal{A}}(\boldsymbol{s},\boldsymbol{a}) - r - \gamma\bar{Q}^{\mathcal{A}}(\boldsymbol{s}',\boldsymbol{a}'))^2\right]$
  Update $Q^{\mathcal{W}}$: $\min_{Q^{\mathcal{W}}} \mathbb{E}_{\boldsymbol{s}\sim\mathcal{B},\boldsymbol{w}\sim\mathcal{N}(0,I)}[(Q^{\mathcal{W}}(\boldsymbol{s},\boldsymbol{w}) - Q^{\mathcal{A}}(\boldsymbol{s}, \pi_{\text{dp}}^{\mathcal{W}}(\boldsymbol{s},\boldsymbol{w})))^2]$
  Update $\pi^{\mathcal{W}}$: $\max_{\pi^{\mathcal{W}}} \mathbb{E}_{\boldsymbol{s}\sim\mathcal{B}}\left[Q^{\mathcal{W}}(\boldsymbol{s}, \pi^{\mathcal{W}}(\boldsymbol{s}))\right]$
  if access to online environment $\mathcal{M}$ **then**
    Sample latent-noise action $\boldsymbol{w}_t \sim \pi^{\mathcal{W}}(\boldsymbol{s}_t)$ and compute $\boldsymbol{a}_t \leftarrow \pi_{\text{dp}}^{\mathcal{W}}(\boldsymbol{s}_t, \boldsymbol{w}_t)$
    Play $\boldsymbol{a}_t$ in $\mathcal{M}$, observe r_t and next state $\boldsymbol{s}_{t+1}$, and add $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ to $\mathcal{B}$

Experiments

Online adaptation

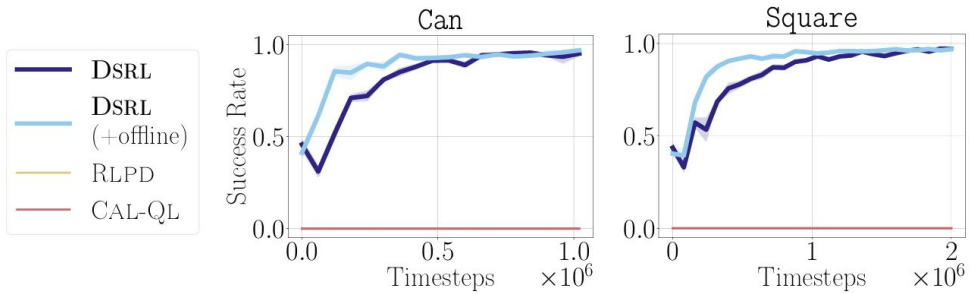


Experiments

Offline adaptation

Task	BC ($\mathcal{N}$)	BC (π_{dp})	IQL	REBRAC	IDQL	SRPO	CAC	FAWAC	FBRAC	IFQL	FQL	DSRL
antmaze-large-navigate-singletask	0 ±0	0 ±0	48 ±9	**91** ±10	0 ±0	0 ±0	42 ±7	1 ±1	70 ±20	24 ±17	80 ±8	40 ±29
antmaze-giant-navigate-singletask	0 ±0	0 ±0	0 ±0	**27** ±22	0 ±0	0 ±0	0 ±0	0 ±0	0 ±1	0 ±0	4 ±5	0 ±0
humanoidmaze-medium-navigate-singletask	1 ±0	0 ±0	32 ±7	16 ±9	1 ±1	0 ±0	38 ±19	6 ±2	25 ±8	**69** ±19	19 ±12	34 ±20
humanoidmaze-large-navigate-singletask	0 ±0	0 ±0	3 ±1	2 ±1	0 ±0	0 ±0	1 ±1	0 ±0	0 ±1	6 ±2	7 ±6	**10** ±12
antsoccer-arena-navigate-singletask	1 ±0	0 ±0	3 ±2	0 ±0	0 ±1	0 ±0	0 ±0	12 ±3	24 ±4	16 ±9	**39** ±6	28 ±9
cube-single-play-singletask	3 ±1	3 ±3	85 ±8	92 ±4	**96** ±2	82 ±16	80 ±30	81 ±9	83 ±13	73 ±3	**97** ±2	**93** ±14
cube-double-play-singletask	0 ±0	0 ±0	1 ±1	7 ±3	16 ±10	0 ±0	2 ±2	2 ±1	22 ±12	9 ±5	36 ±6	**53** ±14
scene-play-singletask	1 ±1	0 ±0	12 ±3	50 ±13	33 ±14	2 ±2	50 ±40	18 ±8	46 ±10	0 ±0	76 ±9	**88** ±9
puzzle-3x3-play-singletask	1 ±1	0 ±0	2 ±1	2 ±1	0 ±0	0 ±0	0 ±0	1 ±1	2 ±2	0 ±0	**16** ±5	0 ±0
puzzle-4x4-play-singletask	0 ±0	0 ±0	5 ±2	10 ±3	26 ±6	7 ±4	1 ±1	0 ±0	5 ±1	21 ±11	11 ±3	**37** ±13

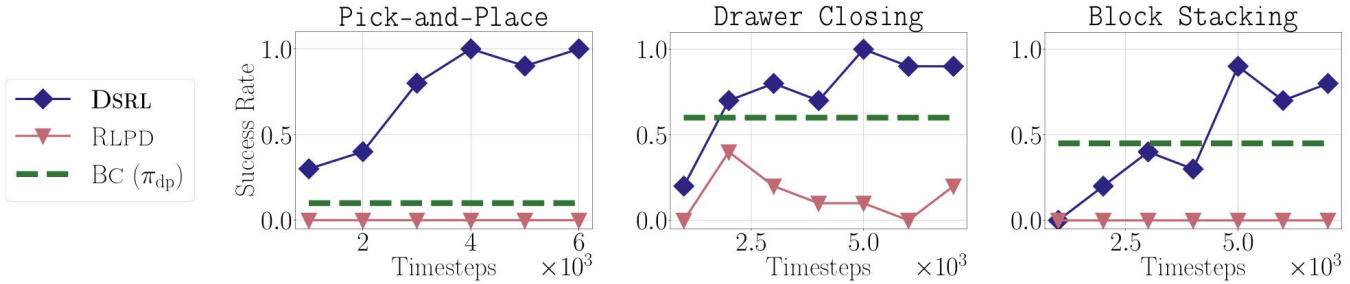
Offline-to-online adaptation



Single-task adaptation in real-world

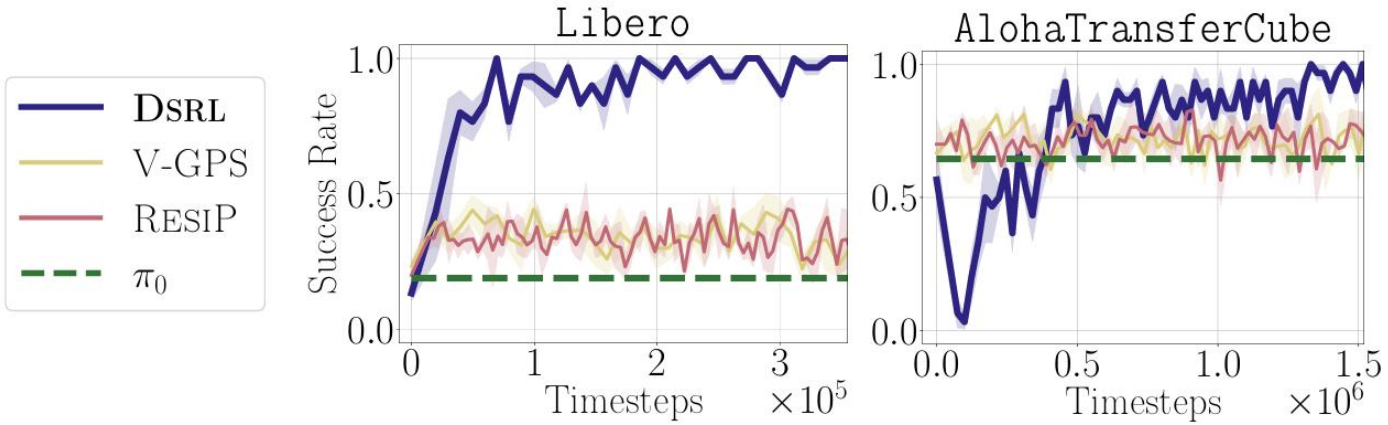
π_{dp}	RLPD	RLPD + interventions	DSRL
2/10	0/10	0/10	**9/10**

Multi-task adaptation in real-world



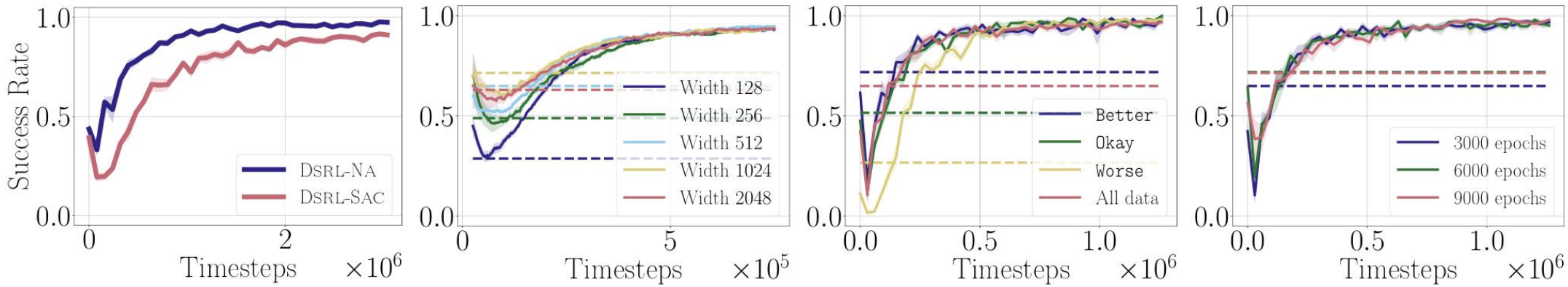
Experiments

VLA adaptation



Task	π_0	DSRL
Turn on toaster	5/20	**18/20**
Put spoon on plate	15/20	**19/20**

Ablation study

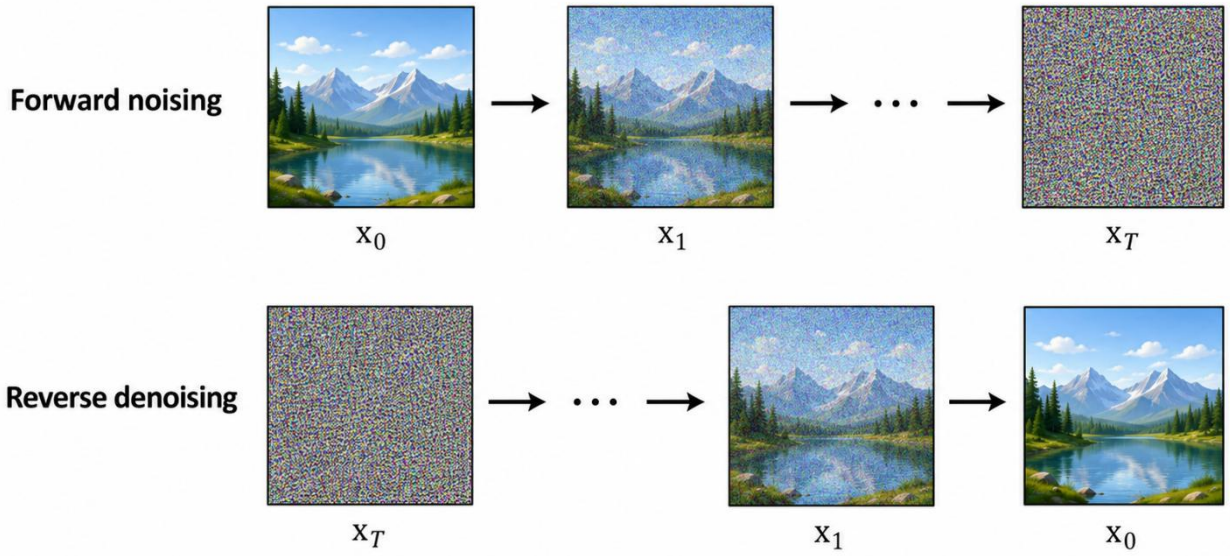


Flow Q-Learning

Seohong Park [1] Qiyang Li [1] Sergey Levine [1]

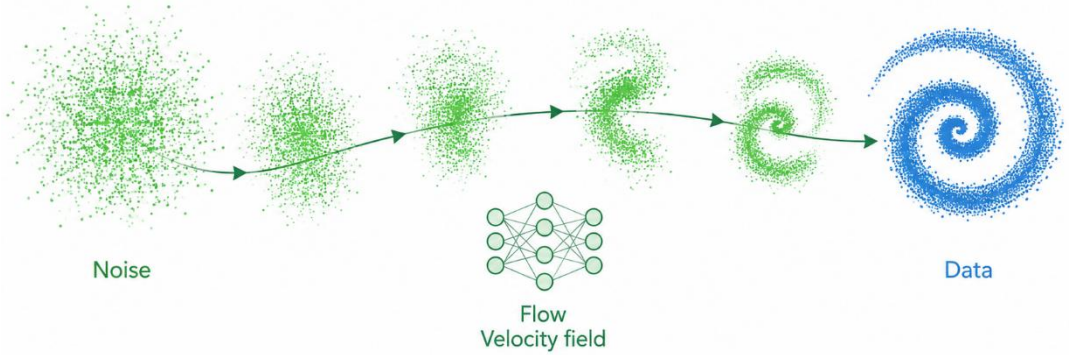
Background

Diffusion Model



Learning objective: reverse denoising process

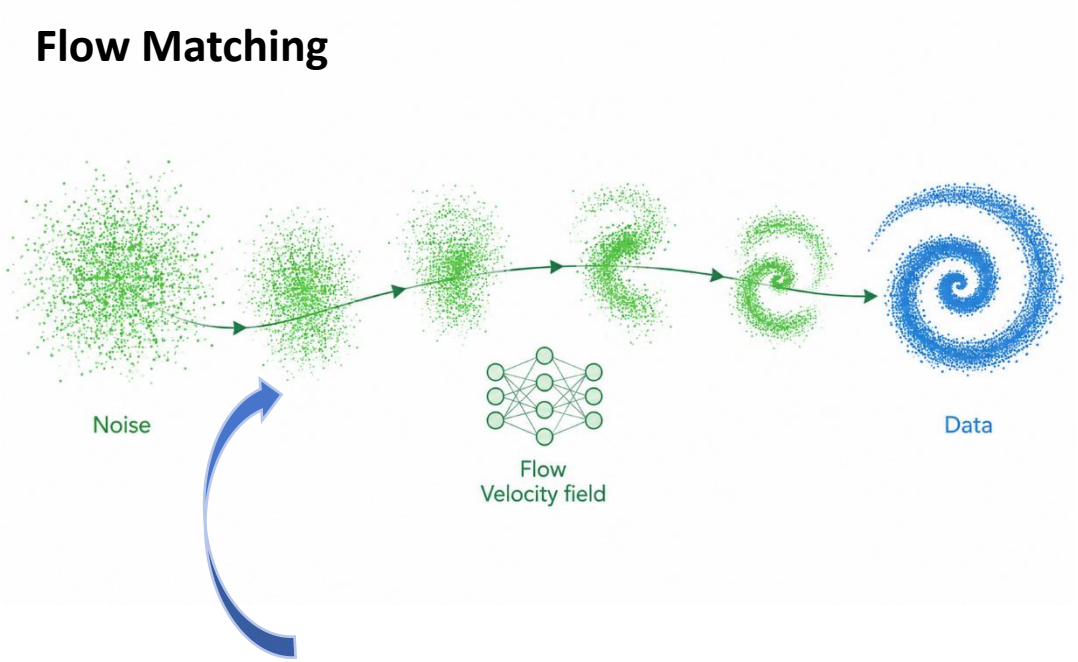
Flow Matching



Learning objective:
continuous flow from noise to data (velocity field)

Background

Flow Matching



distribution at time t

$$\frac{dx(t)}{dt} = v_\theta(x(t), t)$$

velocity field

$$x_0 \sim p_0$$

$$x_1 \sim p_{\text{data}}$$

$$x_1 = x_0 + \int_0^1 v_\theta(t, x_t)\, dt$$

Intermediate distribution $\{p_t(x)\}_{t\in[0,1]}$

Velocity field $\frac{dx_t}{dt} = u_t(x_t)$

Continuity condition $\frac{\partial p_t(x)}{\partial t} + \nabla \cdot (p_t(x)u_t(x)) = 0$

Approximation $v_\theta(t, x) \approx u_t(x)$

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t\sim U[0,1],\, x\sim p_t}\left[\|v_\theta(t, x) - u_t(x)\|^2\right]$$

Conditional flow matching

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,x_0,x_1}\left[\|v_\theta(t, x_t) - u_t(x_t|x_0, x_1)\|^2\right]$$

Rectified flow

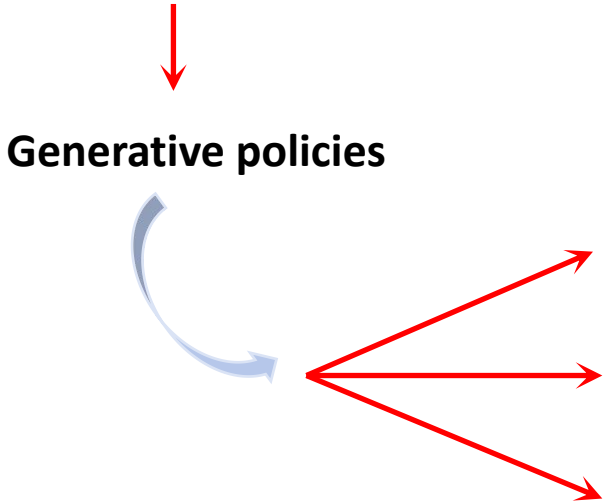
$$x_t = (1-t)x_0 + tx_1$$

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,x_0,x_1}\left[\|v_\theta\left(t, (1-t)x_0 + tx_1\right) - (x_1 - x_0)\|^2\right]$$

Motivation

Problems in offline RL

- Limited expressiveness of Gaussian policies



Generative policies

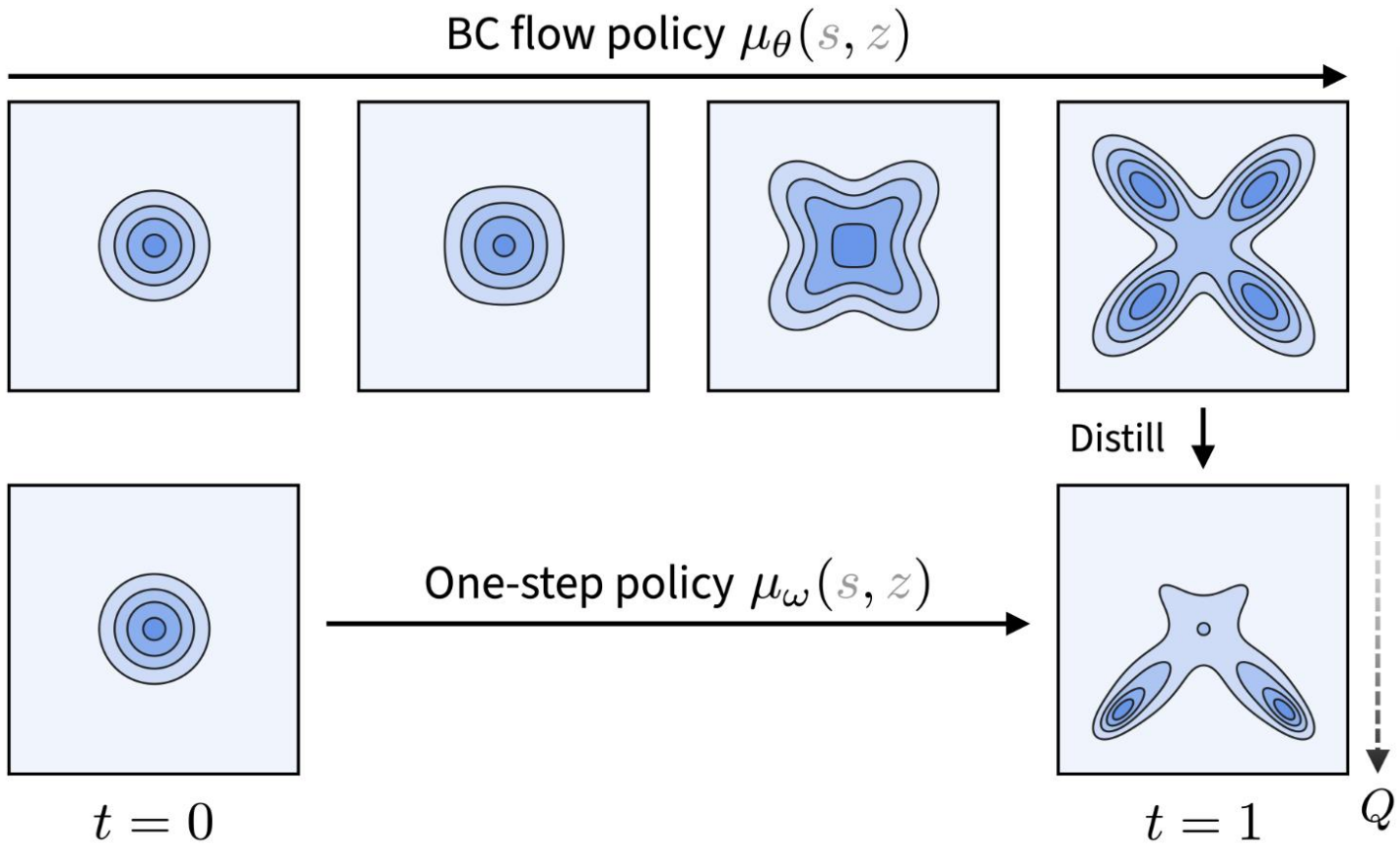
Only imitate dataset actions but cannot further optimize task returns

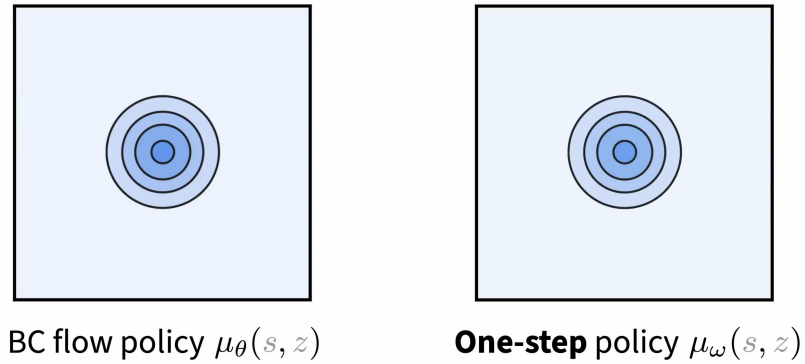
Directly training with RL leads to unstable optimization and high cost

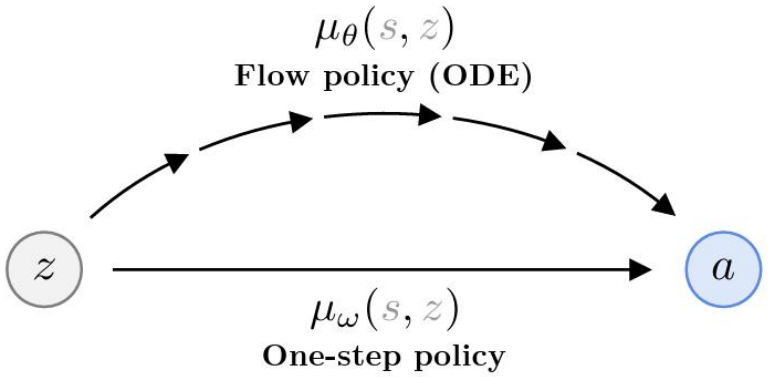
Difficult to seamlessly integrate into existing algorithms

How to train an expressive policy that can be integrated with existing RL algorithms?

Method







*Use flow matching to model **complex action distributions**, and use Q-learning to guide a one-step policy toward **higher returns**.*

Method

Algorithm 1 Flow Q-Learning (FQL)

function $\mu_\theta(s, z)$ ▷ BC flow policy
  for $t = 0, 1, \ldots, M-1$ **do**
    $z \leftarrow z + v_\theta(t/M, s, z)/M$ ▷ Euler method
  return z

while not converged **do**
  Sample batch $\{(s, a, r, s')\} \sim \mathcal{D}$
  ▷ Train critic Q_ϕ
  $z \sim \mathcal{N}(0, I_d)$
  $a' \leftarrow \mu_\omega(s', z)$
  Update ϕ to minimize $\mathbb{E}[(Q_\phi(s, a) - r - \gamma Q_{\bar{\phi}}(s', a'))^2]$
  ▷ Train vector field v_θ in BC flow policy π_θ
  $x^0 \sim \mathcal{N}(0, I_d)$
  $x^1 \leftarrow a$
  $t \sim \mathrm{Unif}([0, 1])$
  $x^t \leftarrow (1-t)x^0 + tx^1$
  Update θ to minimize $\mathbb{E}[\|v_\theta(t, s, x^t) - (x^1 - x^0)\|_2^2]$
  ▷ Train one-step policy π_ω
  $z \sim \mathcal{N}(0, I_d)$
  $a^\pi \leftarrow \mu_\omega(s, z)$
  Update ω to minimize $\mathbb{E}[-Q_\phi(s, a^\pi) + \alpha\|a^\pi - \mu_\theta(s, z)\|_2^2]$
return One-step policy π_ω

$$\mathcal{L}_{\text{Distill}}(\omega) = \mathbb{E}_{\substack{s\sim\mathcal{D},\\ z\sim\mathcal{N}(0,I_d)}} \left[\|\mu_\omega(s, z) - \mu_\theta(s, z)\|_2^2\right]$$

noise $z \sim \mathcal{N}(0, I)$ state s

iteration *MLP net*

BC flow policy $\mu_\theta(s, z)$ one-step policy $\mu_\omega(s, z)$

$$\begin{aligned}\mathcal{L}_{\text{Distill}}(\omega) &= \mathbb{E}_{\substack{s\sim\mathcal{D},\\ z\sim\mathcal{N}(0,I_d)}} \left[\|\mu_\omega(s, z) - \mu_\theta(s, z)\|_2^2\right] \\ &\geq \mathbb{E}_{s\sim\mathcal{D}} \left[\inf_{\lambda\in\Lambda(\pi_\omega,\pi_\theta)} \mathbb{E}_{x,y\sim\lambda}[\|x - y\|_2^2]\right] \\ &= \mathbb{E}_{s\sim\mathcal{D}} \left[W_2(\pi_\omega, \pi_\theta)^2\right]\end{aligned}$$

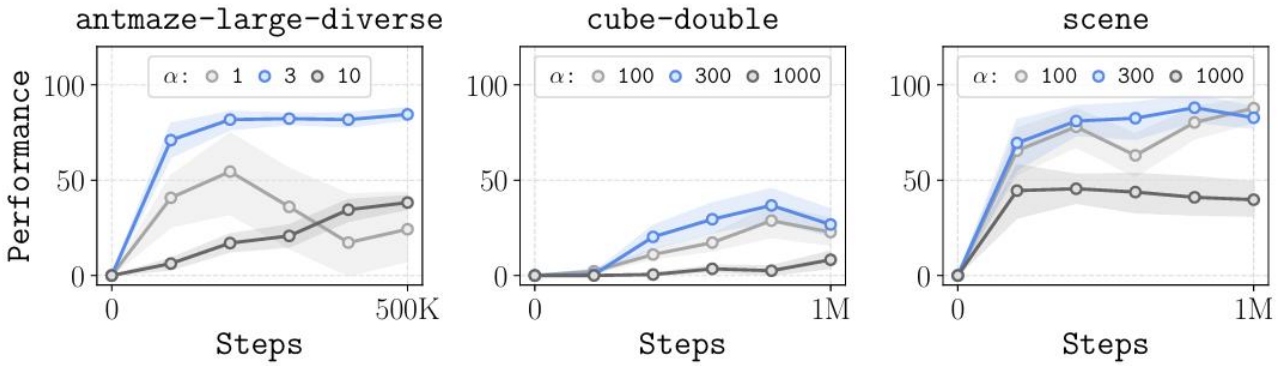
Experiments

Offline performance

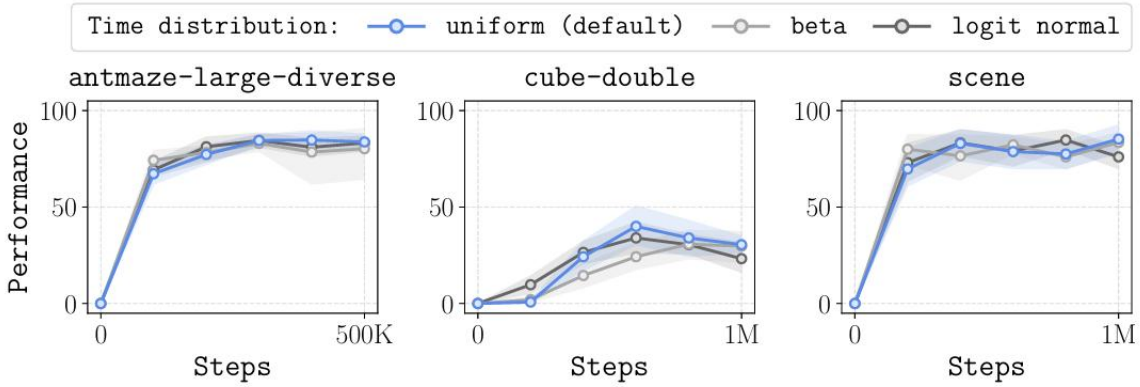
	Gaussian Policies			Diffusion Policies			Flow Policies			
Task Category	BC	IQL	ReBRAC	IDQL	SRPO	CAC	FAWAC	FBRAC	IFQL	FQL
OGBench antmaze-large-singletask (5 tasks)	11 ±1	53 ±3	**81** ±5	21 ±5	11 ±4	33 ±4	6 ±1	60 ±6	28 ±5	**79** ±3
OGBench antmaze-giant-singletask (5 tasks)	0 ±0	4 ±1	**26** ±8	0 ±0	0 ±0	0 ±0	0 ±0	4 ±4	3 ±2	9 ±6
OGBench humanoidmaze-medium-singletask (5 tasks)	2 ±1	33 ±2	22 ±8	1 ±0	1 ±1	53 ±8	19 ±1	38 ±5	**60** ±14	**58** ±5
OGBench humanoidmaze-large-singletask (5 tasks)	1 ±0	2 ±1	2 ±1	1 ±0	0 ±0	0 ±0	0 ±0	2 ±0	**11** ±2	4 ±2
OGBench antsoccer-arena-singletask (5 tasks)	1 ±0	8 ±2	0 ±0	12 ±4	1 ±0	2 ±4	12 ±0	16 ±1	33 ±6	**60** ±2
OGBench cube-single-singletask (5 tasks)	5 ±1	83 ±3	91 ±2	**95** ±2	80 ±5	85 ±9	81 ±4	79 ±7	79 ±2	**96** ±1
OGBench cube-double-singletask (5 tasks)	2 ±1	7 ±1	12 ±1	15 ±6	2 ±1	6 ±2	5 ±2	15 ±3	14 ±3	**29** ±2
OGBench scene-singletask (5 tasks)	5 ±1	28 ±1	41 ±3	46 ±3	20 ±1	40 ±7	30 ±3	45 ±5	30 ±3	**56** ±2
OGBench puzzle-3x3-singletask (5 tasks)	2 ±0	9 ±1	21 ±1	10 ±2	18 ±1	19 ±0	6 ±2	14 ±4	19 ±1	**30** ±1
OGBench puzzle-4x4-singletask (5 tasks)	0 ±0	7 ±1	14 ±1	**29** ±3	10 ±3	15 ±3	1 ±0	13 ±1	25 ±5	17 ±2
D4RL antmaze (6 tasks)	17	57	78	79	74	30 ±3	44 ±3	64 ±7	65 ±7	**84** ±3
D4RL adroit (12 tasks)	48	53	**59**	52 ±1	51 ±1	43 ±2	48 ±1	50 ±2	52 ±1	52 ±1
Visual manipulation (5 tasks)	-	42 ±4	60 ±2	-	-	-	-	22 ±2	50 ±5	**65** ±2

Experiments

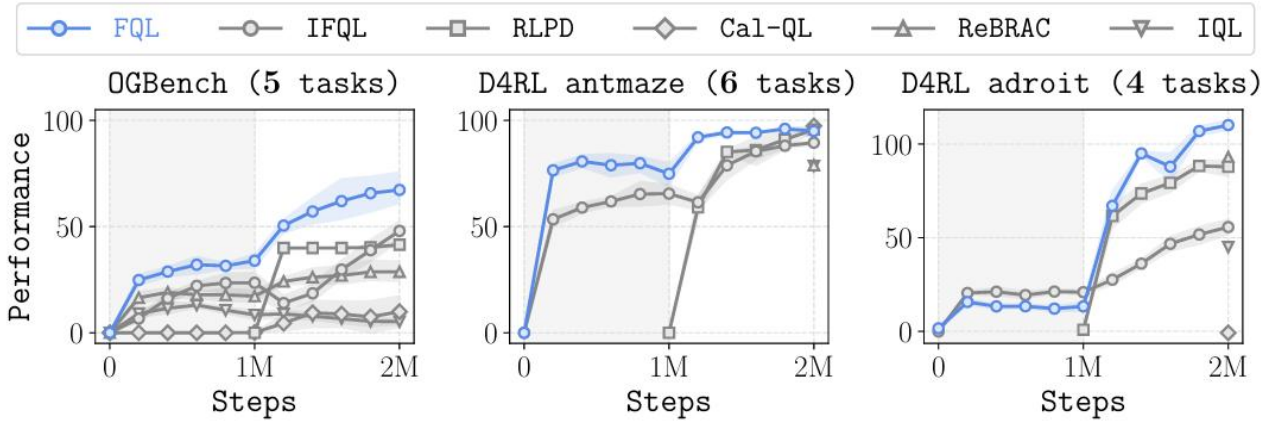
Hyperparameter sensitivity



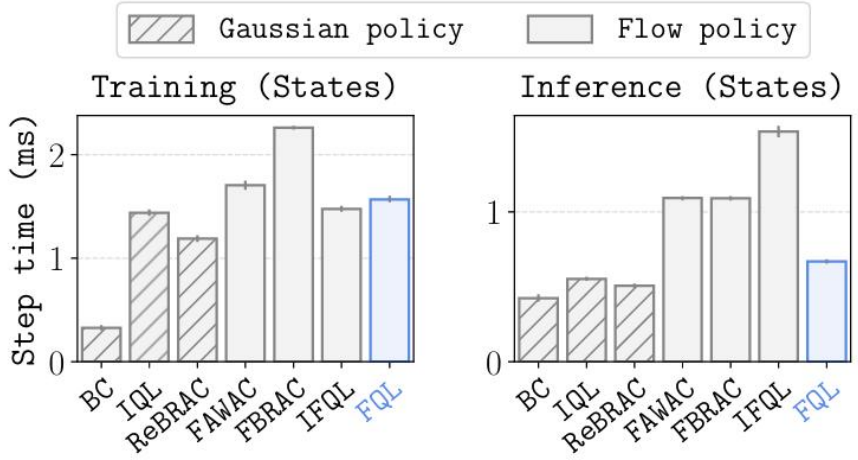
Time distribution



Offline-to-online performance



Time cost



Thanks