Q-learning with Adjoint Matching

Qiyang Li, Sergey Levine
UC Berkeley

code

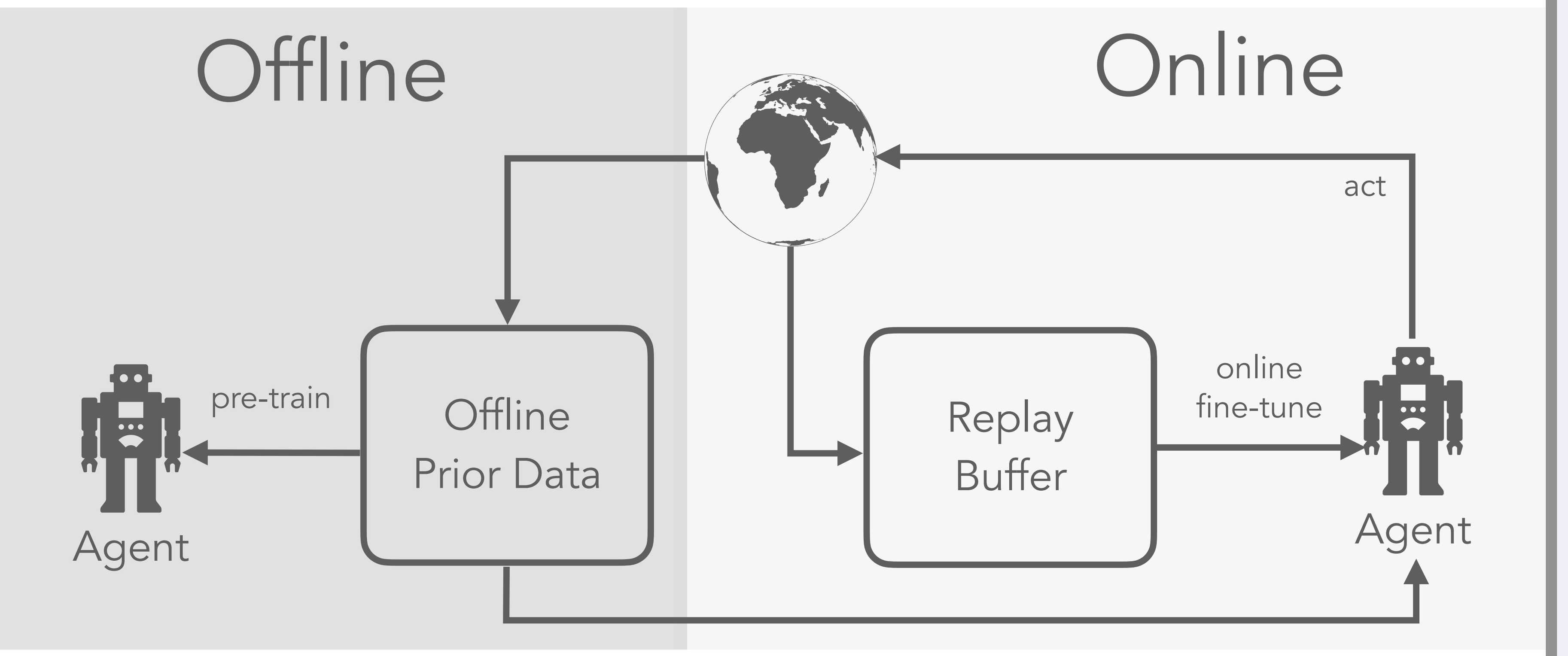
website

Setup and Challenges

Given: offline prior data (e.g., expert, exploratory data)

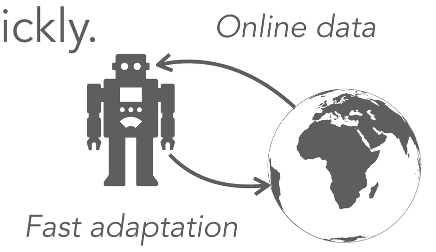
Goal: sample-efficient *offline-to-online RL*

1. **Offline —** Given offline prior data, learn a policy.
2. **Online —** Fine-tune the policy with online interactions.



Challenge 1: Prior data is multi-modal and complex.

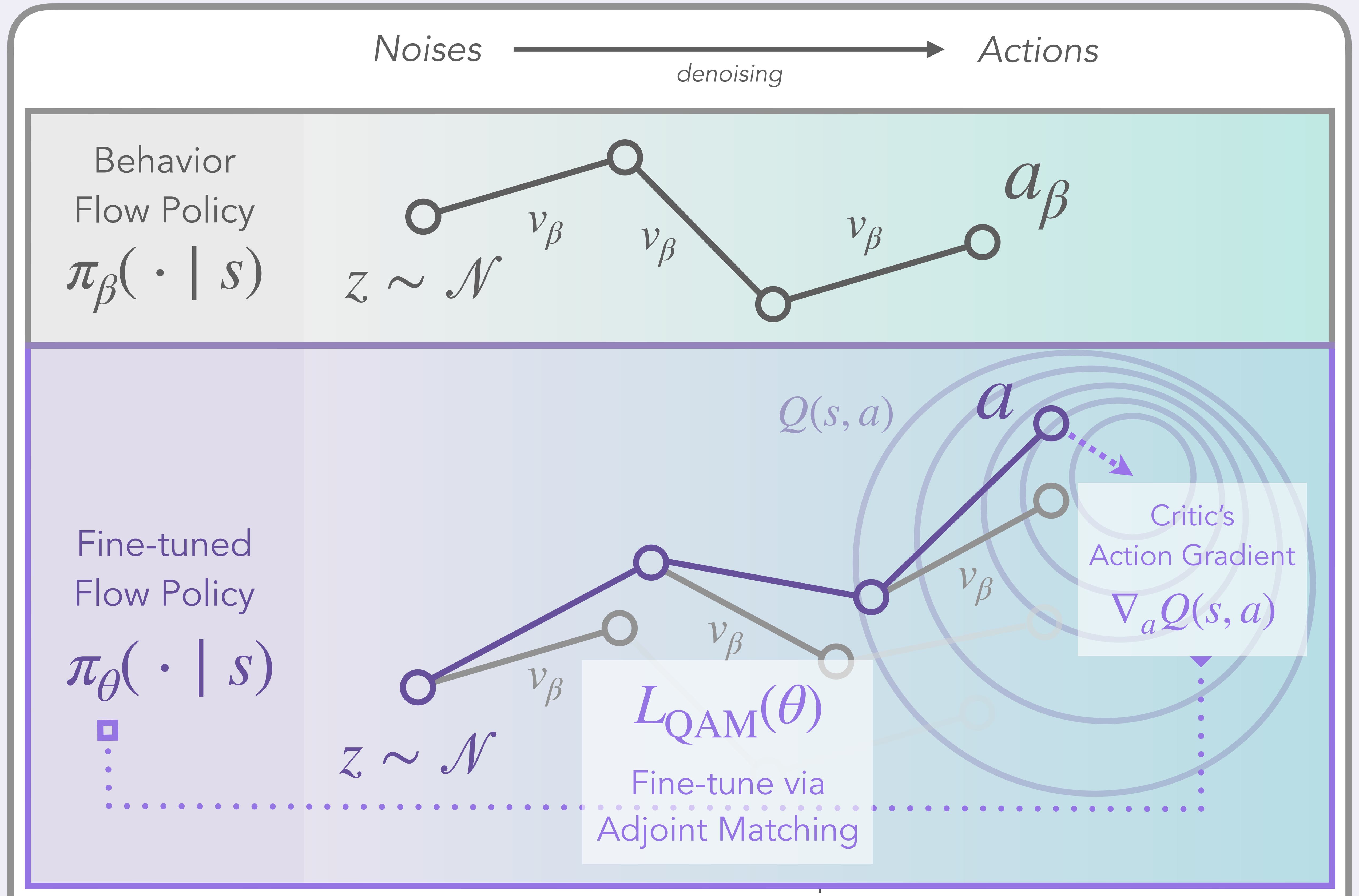
Challenge 2: RL updates must be efficient and adapt to online data quickly.



Key Ingredients of Efficient RL Algorithm

1. Use critic's action gradient $\nabla_a Q(s, a)$ to train policy
2. Use expressive policy (e.g., flow/diffusion)
3. Stable optimization (e.g., no naïve backpropagation through time)

	Action Gradient ($\nabla_a Q(s,a)$)	No BPTT	Optimum ($\propto \pi_\beta e^Q$)	Multi-step Flow/Diffusion
ReBRAC	○	○		
FBRAC	○			○
BAM	○		○	○
FQL	○	○		
FAWAC		○	○	○
QSM/DAC/CGQL*	○	○		○
DSRL	○	○		○
FEdit	○	○		○
IFQL		○		○
QAM (ours)	○	○	○	○



Adjoint Matching (Domingo-Enrich et al., 2025)

We adapted it to Q-learning!

Step 1, SDE: $\mathrm{d}a_t = \left(2v(a_t, t) - a_t/t\right)\mathrm{d}t + \sqrt{2(1-t)/t}\,\mathrm{d}W_t$

Step 2, Adjoint: $\mathrm{d}\tilde{g}(\mathbf{a}, t) = -\nabla_{a_t}\left[2v_\beta(a_t, t) - a_t/t\right]\tilde{g}(\mathbf{a}, t)\mathrm{d}t$

boundary condition — $\tilde{g}(\mathbf{a}, 1) = -\nabla_{a_1} Q(s, a_1)$

Step 3, Matching! $L_{\mathrm{AM}}(\theta) = \mathbb{E}_{\mathbf{a}\sim\mathrm{SDE}}\left[\int_0^1 \|2(v_\theta(a_t, t) - v_\beta(a_t, t))/\sigma_t + \sigma_t \tilde{g}(\mathbf{a}, t)\|_2^2 \mathrm{d}t\right]$

$\sigma_t = \sqrt{2(1-t)/t}$

Two residual RL variants

QAM-E = QAM + Gaussian edit policy

$a \sim \pi^\theta(\cdot \mid s)$ *QAM fine-tuned policy*

$\Delta_a \sim \pi^\omega(\cdot \mid s, a)$ *Edit policy*

QAM-F = QAM + FQL

$$L_{\mathrm{FQL}}(\omega) = -\mathbb{E}_{\mu^\omega}[Q(a_1)] + \mathbb{E}_D\left[\left\| \mu^\omega(a_0, t) - \int_0^1 v_t^\theta(a_t)\mathrm{d}t - a_0 \right\|_2^2\right]$$

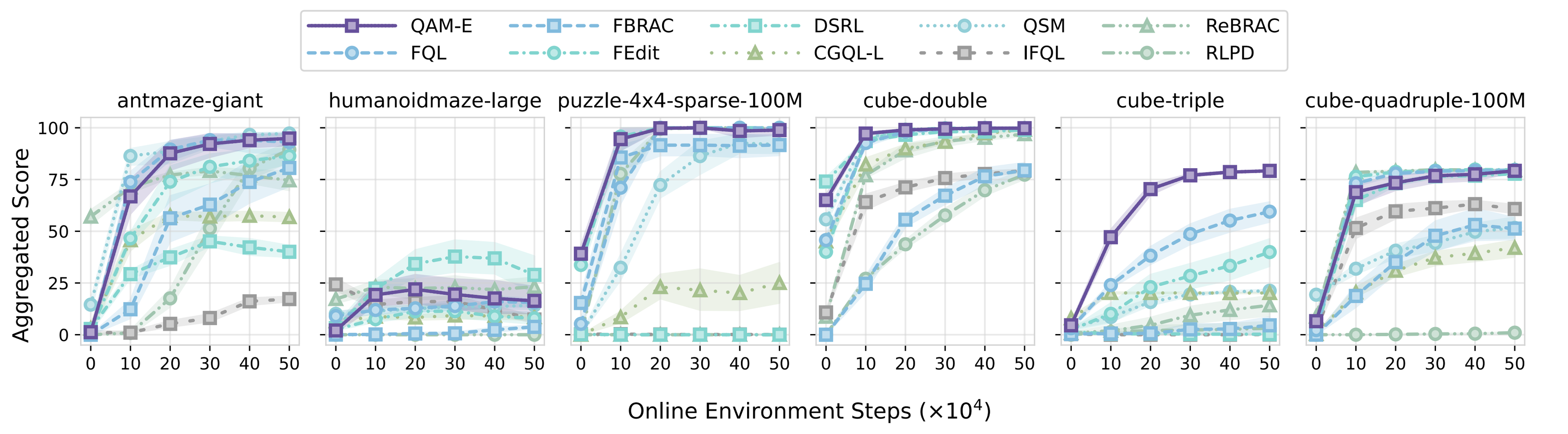
Results

(I) How does it compare to prior methods?

Offline RL

		al (5 tasks)	ag (5 tasks)	hm (5 tasks)	hl (5 tasks)	scene (5 tasks)	p33 (5 tasks)	p44 (5 tasks)	c2 (5 tasks)	c3 (5 tasks)	c4 (5 tasks)	all (50 tasks)
Gaussian	ReBRAC	[94,95] **94**	[53,60] **57**	[65,74] 69	[15,19] 17	[61,69] 65	[73,84] 79	[0,0] 0	[8,10] 9	[0,1] 1	[6,11] 9	[39,41] 40
Backprop	FBRAC	[1,4] 2	[0,0] 0	[37,41] 39	[0,0] 0	[42,57] 50	[0,1] 0	[11,19] 15	[0,0] 0	[0,1] 0	[0,0] 0	[10,12] 11
	BAM	[82,85] 84	[0,2] 1	[58,62] 60	[4,7] 5	[98,99] **98**	[48,64] 56	[0,0] 0	[44,50] 47	[2,5] 3	[0,0] 0	[34,36] 35
	FQL	[72,79] 76	[0,0] 0	[63,74] 68	[7,11] 9	[77,80] 78	[60,78] 70	[3,8] 5	[43,49] 46	[2,4] 3	[1,5] 2	[34,37] 36
Adv. Weighted	FAWAC	[15,19] 17	[0,0] 0	[22,26] 24	[0,0] 0	[35,41] 38	[2,3] 3	[0,0] 0	[2,2] 2	[0,0] 0	[0,0] 0	[8,9] 8
Guidance	CGQL	[73,80] 76	[0,2] 0	[57,62] 60	[4,5] 5	[36,40] 38	[39,55] 48	[16,33] 24	[36,41] 38	[7,9] **8**	[0,0] 0	[28,31] 30
	CGQL-M	[68,73] 71	[1,8] 4	[40,43] 42	[3,8] 6	[72,76] 74	[100,100] **100**	[0,0] 0	[39,43] 41	[7,9] **8**	[0,1] 1	[34,35] 35
	CGQL-L	[62,67] 65	[1,6] 3	[57,67] 62	[5,8] 6	[85,91] 88	[83,96] 90	[0,0] 0	[43,47] 45	[7,9] **8**	[0,1] 0	[36,38] 37
	DAC	[86,90] 88	[11,20] 16	[81,85] **83**	[0,0] 0	[65,70] 68	[62,75] 68	[0,0] 0	[33,36] 35	[3,6] **5**	[1,5] 3	[35,38] 36
	QSM	[89,93] 91	[12,17] 15	[80,86] **83**	[9,11] 10	[84,87] 86	[49,57] 53	[0,0] 0	[53,58] 56	[3,4] 3	[19,20] **19**	[41,42] 42
Post-processing	DSRL	[56,66] 61	[1,4] 3	[48,57] 53	[2,5] 3	[99,100] **99**	[82,92] 87	[0,0] 0	[72,76] **74**	[1,2] 1	[2,3] 2	[38,39] 38
	FEdit	[54,62] 58	[1,3] 2	[20,23] 22	[2,3] 3	[60,65] 62	[98,100] 99	[30,37] **34**	[37,43] 40	[2,3] 2	[3,7] 5	[32,33] 33
	IFQL	[32,39] 36	[0,2] 1	[85,87] **86**	[21,27] **24**	[83,85] 84	[100,100] **100**	[0,0] 0	[10,12] 11	[0,0] 0	[1,3] 2	[34,35] 34
Adjoint Matching	QAM	[78,84] 81	[14,22] 18	[64,69] 67	[9,14] 11	[96,98] 97	[99,100] 100	[0,0] 0	[62,66] 64	[3,4] 3	[2,4] 3	[44,45] **44**
	QAM-F	[81,84] 83	[8,16] 12	[62,68] 65	[9,14] 12	[92,97] 95	[99,100] 99	[5,8] 6	[63,67] 65	[2,3] 3	[12,17] 14	[45,46] **45**
	QAM-E	[80,86] 83	[0,3] 1	[54,63] 59	[1,3] 2	[96,98] 97	[100,100] **100**	[35,43] **39**	[63,68] 65	[4,6] 5	[4,9] 6	[45,47] **46**

Online Fine-tuning



(II) Hyperparameter Sensitivity