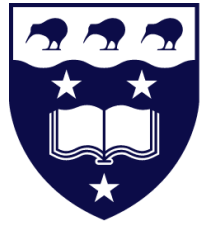
Loop-Free Inverse Reinforcement Learning

via Sequential Value Recovery with Q-Score Matching

Yitan Zhang

NeurIPS 2026



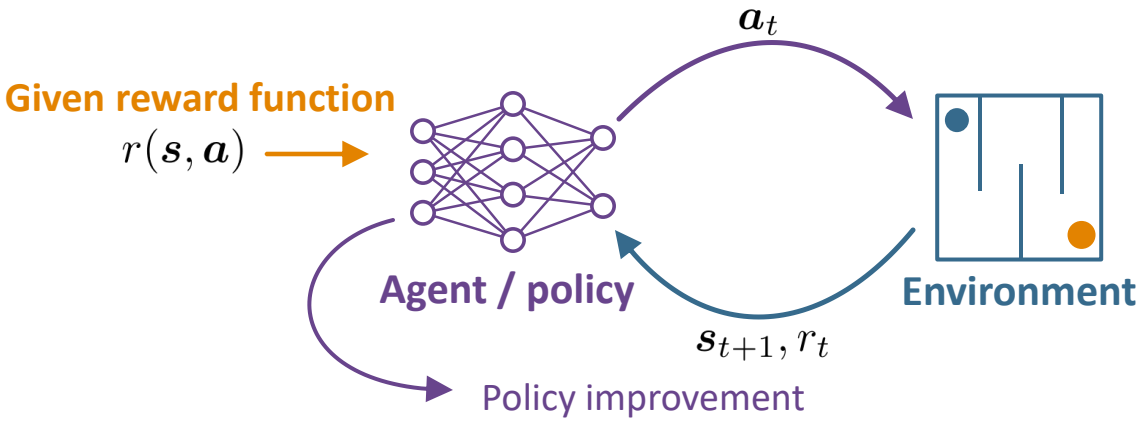


Waipapa
Taumata Rau
University
of Auckland



Reinforcement Learning and Inverse RL

Reinforcement Learning: from reward to behavior

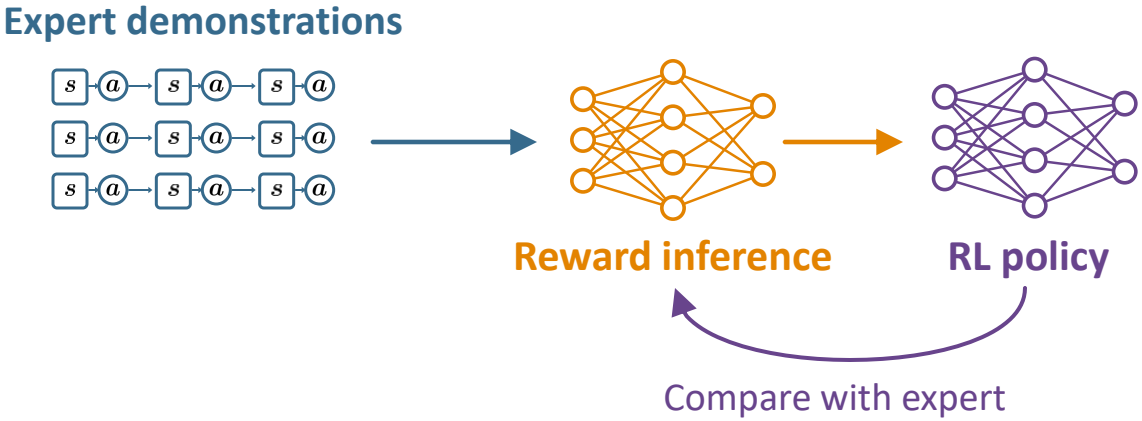


Training target

Choose actions that maximize expected discounted return.

$$\pi^* \in \arg\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t)\right]$$

Inverse Reinforcement Learning: from expert behavior to reward



Training target

Infer a reward that makes the expert policy optimal under the chosen RL model.

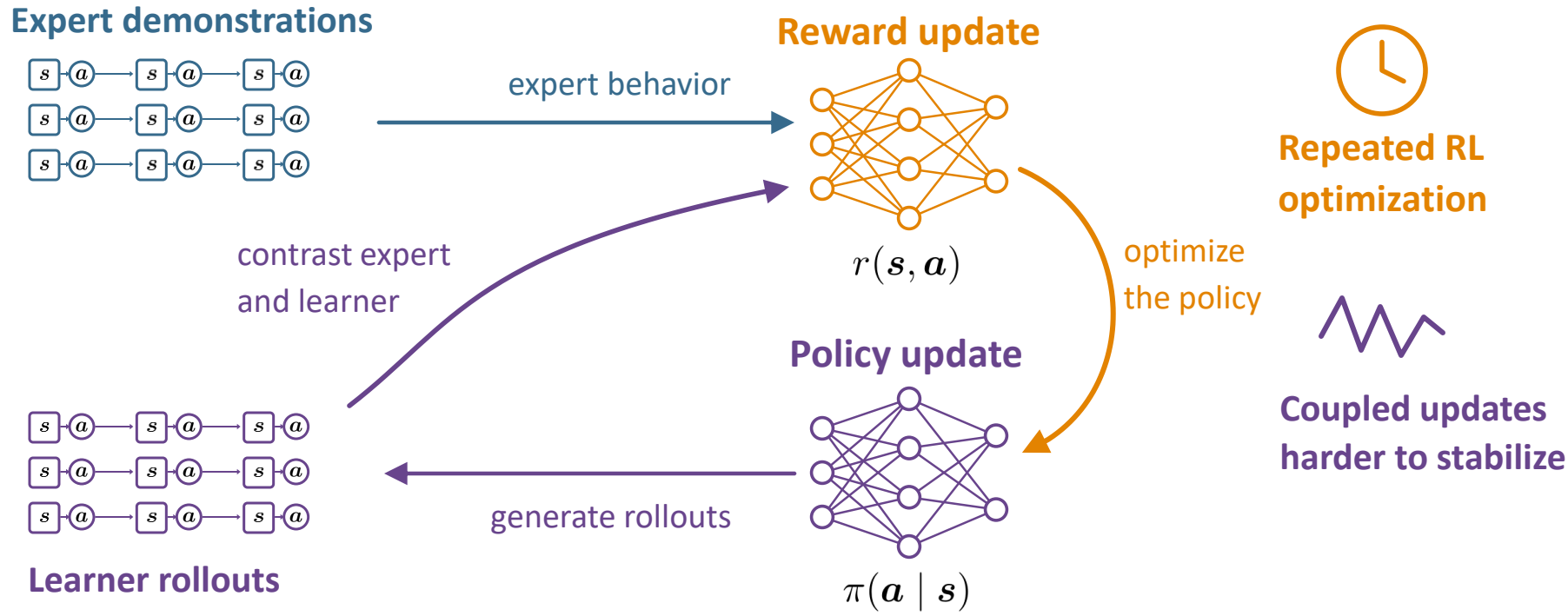
$$\text{Find } r \text{ such that } \pi_E \in \arg\max_{\pi} \mathbb{E}_{\rho_\pi}[r] + \varepsilon H_\mu(\pi)$$

RL: Sutton & Barto, Ch. 3. IRL: Ng & Russell (2000); paper Sec. 3.1.

The Reward–Policy Bottleneck

Reward learning depends on a repeatedly updated learner

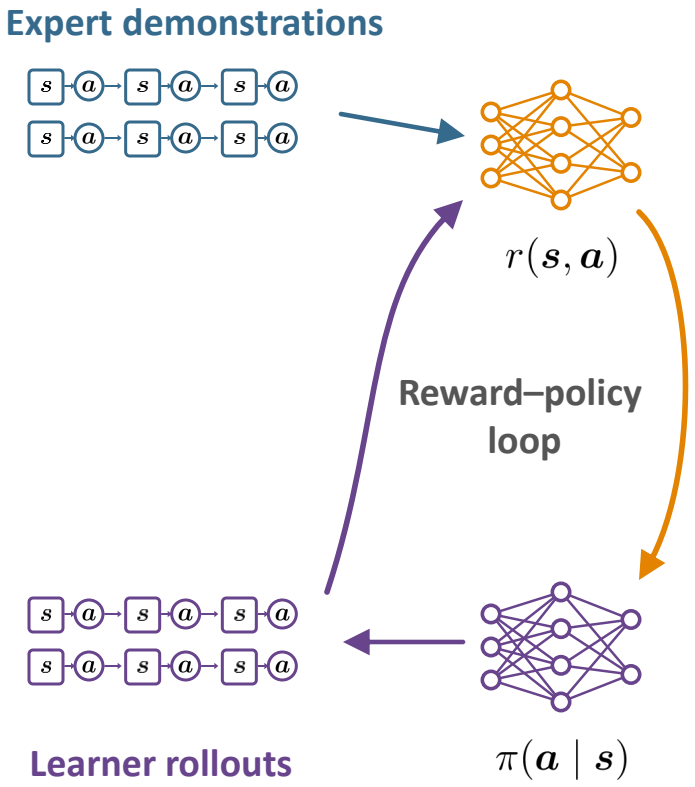
$$\min_{r} \max_{\pi} \quad \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\rho_\pi}[r(\boldsymbol{s},\boldsymbol{a})] + \varepsilon H_\mu(\pi) - \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\rho_{\pi_E}}[r(\boldsymbol{s},\boldsymbol{a})]$$



Can we recover a useful reward without the reward–policy loop?

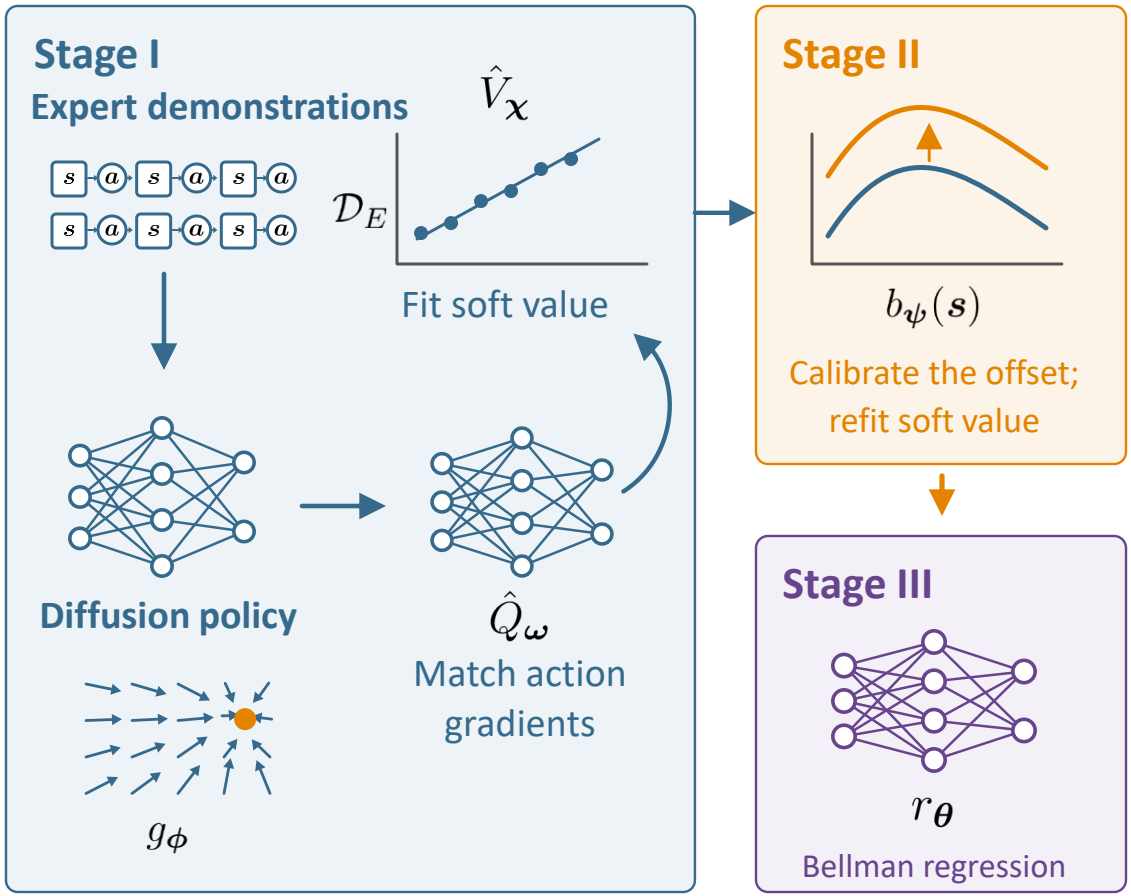
A Sequential, Fully Offline Pipeline

Traditional IRL



Repeated policy optimization

Loop-Free IRL



No reward–policy loop

Inspiration: Diffusion Scores Supervise Values

Expert action directions can directly supervise soft values

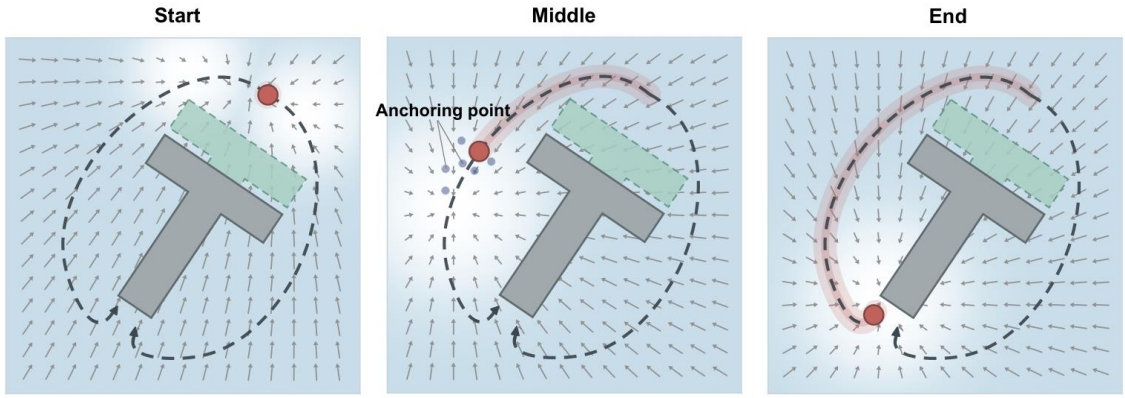


Figure 2: local action preferences in Push-T

A diffusion policy learns to denoise expert actions.

Its learned score:

$$g_{\phi}(\boldsymbol{a}_t^k, \boldsymbol{s}_t, k) := -\frac{\boldsymbol{\epsilon}_{\phi}(\boldsymbol{a}_t^k, \boldsymbol{s}_t, k)}{\sqrt{1-\bar{\alpha}_k}}$$

With a soft-optimal expert and an exact score (Sec. 3.2):

$$g_{\phi}(\boldsymbol{a}_t^k, \boldsymbol{s}_t, k) = \nabla_{\boldsymbol{a}} \log \mu(\boldsymbol{a} \mid \boldsymbol{s}_t)\big|_{\boldsymbol{a}=\boldsymbol{a}_t^k} + \frac{1}{\varepsilon} \nabla_{\boldsymbol{a}} Q_r^*(\boldsymbol{s}_t, \boldsymbol{a})\big|_{\boldsymbol{a}=\boldsymbol{a}_t^k}$$

Diffusion score

Action-gradient supervision

$$\mu \text{ uniform}, \ \varepsilon = 1 \quad \Longrightarrow \quad g_{\phi} = \nabla_{\boldsymbol{a}} Q^*$$

Recover soft action values without an iteratively updated learner.

Stage I: Recover the Soft Values

Two complementary terms recover the soft action value

$$\mathcal{L}_{\hat{Q}}(\boldsymbol{\omega}) := \mathbb{E}_{\substack{\boldsymbol{s}\in\mathcal{S},\boldsymbol{a}\in\mathcal{A},\\ k\sim\mathrm{Unif}\{1,\dots,K\}}} \left[\left\| g_{\boldsymbol{\phi}}(\boldsymbol{a}^k, \boldsymbol{s}, k) - \nabla_{\boldsymbol{a}} \log \mu(\boldsymbol{a} \mid \boldsymbol{s})|_{\boldsymbol{a}=\boldsymbol{a}^k} - \frac{1}{\varepsilon} \nabla_{\boldsymbol{a}} \hat{Q}_{\boldsymbol{\omega}}(\boldsymbol{s}, \boldsymbol{a}) \Big|_{\boldsymbol{a}=\boldsymbol{a}^k} \right\|_2^2 \right]$$

Gradient matching: follow the expert score field

$$+ \lambda \mathbb{E}_{\substack{(\boldsymbol{s},\boldsymbol{a})\sim\mathcal{D}_E,\\ \tilde{\boldsymbol{a}}\sim\eta(\cdot|\boldsymbol{a})}} \left[\mathrm{ReLU}\Big(\xi - \big[\hat{Q}_{\boldsymbol{\omega}}(\boldsymbol{s}, \boldsymbol{a}) - \hat{Q}_{\boldsymbol{\omega}}(\boldsymbol{s}, \tilde{\boldsymbol{a}}) \big] \Big) \right] \tag{8}$$

Value anchoring: expert actions outrank nearby perturbations

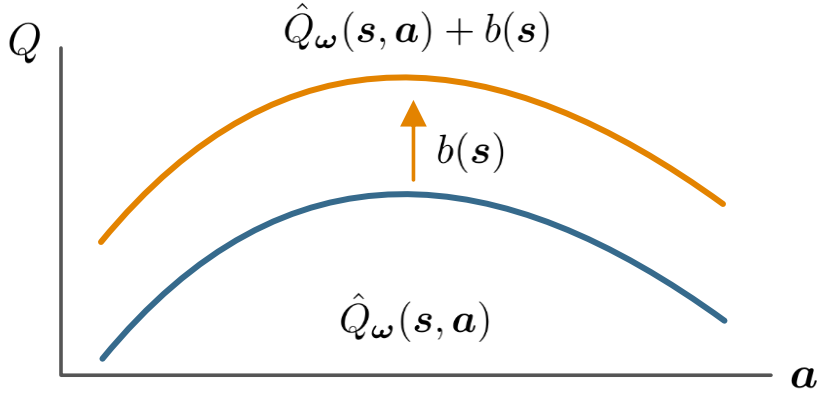
Freeze the recovered action value; then fit the soft value

$$\mathcal{L}_{\hat{V}}(\boldsymbol{\chi}) := \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\mathcal{D}_S} \left[\exp(z) - z - 1 \right], \quad z = \frac{\hat{Q}_{\boldsymbol{\omega}}(\boldsymbol{s}, \boldsymbol{a}) - \hat{V}_{\boldsymbol{\chi}}(\boldsymbol{s})}{\varepsilon} \tag{10}$$

Soft-value fitting via Gumbel regression

Stage II: Calibrate the State Offset

Matching gradients leaves a state-dependent vertical shift



Fixed state · identical action gradients

$$Q^*(\boldsymbol{s},\boldsymbol{a}) = \hat{Q}_{\boldsymbol{\omega}}(\boldsymbol{s},\boldsymbol{a}) + b^*(\boldsymbol{s})$$

The score determines local shape, but cannot determine the offset.

$$\nabla_{\boldsymbol{a}} \hat{Q}_{\boldsymbol{\omega}} = \nabla_{\boldsymbol{a}} (\hat{Q}_{\boldsymbol{\omega}} + b(\boldsymbol{s}))$$

Use transitions to couple offsets across consecutive states:

$$\delta_{\boldsymbol{\omega},\boldsymbol{\chi},\boldsymbol{\psi}}(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1}) := \hat{Q}_{\boldsymbol{\omega}}(\boldsymbol{s}_t,\boldsymbol{a}_t) + b_{\boldsymbol{\psi}}(\boldsymbol{s}_t) - \gamma\left[\hat{V}_{\boldsymbol{\chi}}(\boldsymbol{s}_{t+1}) + b_{\boldsymbol{\psi}}(\boldsymbol{s}_{t+1})\right] \tag{11}$$

$$\mathcal{L}_b(\boldsymbol{\psi}) := \mathbb{E}_{\substack{(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1}) \\ \sim \mathcal{D}_E \cup \mathcal{D}_S}}\left[\delta^2_{\boldsymbol{\omega},\boldsymbol{\chi},\boldsymbol{\psi}} + \lambda_b\, b^2_{\boldsymbol{\psi}}(\boldsymbol{s}_t)\right] + \mathbb{E}_{\boldsymbol{s}_\perp \in \mathcal{S}_\perp, \boldsymbol{a}}\left[\left(\hat{Q}_{\boldsymbol{\omega}}(\boldsymbol{s}_\perp,\boldsymbol{a}) + b_{\boldsymbol{\psi}}(\boldsymbol{s}_\perp)\right)^2\right] \tag{12}$$

Transition-coupled offset regularization

Small-offset penalty

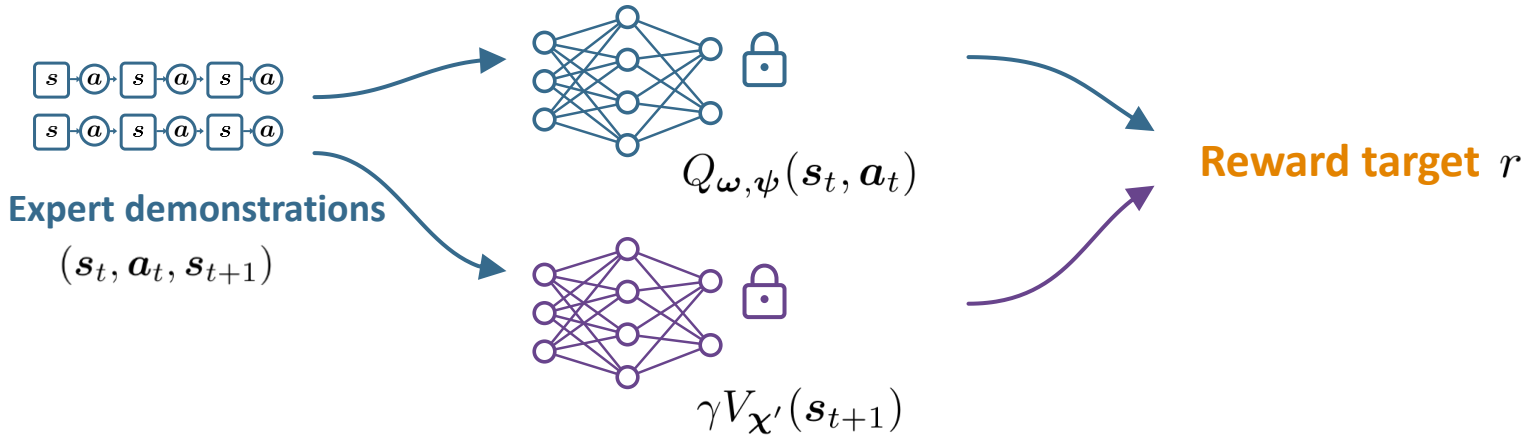
Zero value at absorbing states

Freeze calibrated action values and refit the soft value:

$$\mathcal{L}_V(\boldsymbol{\chi}') := \mathbb{E}_{\boldsymbol{s},\boldsymbol{a}\sim\mu(\cdot|\boldsymbol{s})}\left[\exp(\bar{z}) - \bar{z} - 1\right], \quad \bar{z} := \frac{Q_{\boldsymbol{\omega},\boldsymbol{\psi}}(\boldsymbol{s}_t,\boldsymbol{a}_t) - V_{\boldsymbol{\chi}'}(\boldsymbol{s}_t)}{\varepsilon} \tag{13}$$

Stage III: Recover the Reward

Frozen calibrated values become reward-supervision targets

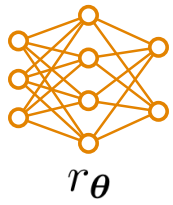


$$r = Q_{\omega,\psi}(s_t, a_t) - \gamma V_{\chi'}(s_{t+1})$$

$$\tilde{r} = \frac{\boxed{\text{clip}(r, -c_r, c_r)} - m_r}{\sigma_r + \zeta_r} \qquad \tilde{r}_{\theta} = \frac{r_{\theta}(s_t, a_t) - m_r}{\sigma_r + \zeta_r} \tag{14}$$

Clip extreme targets

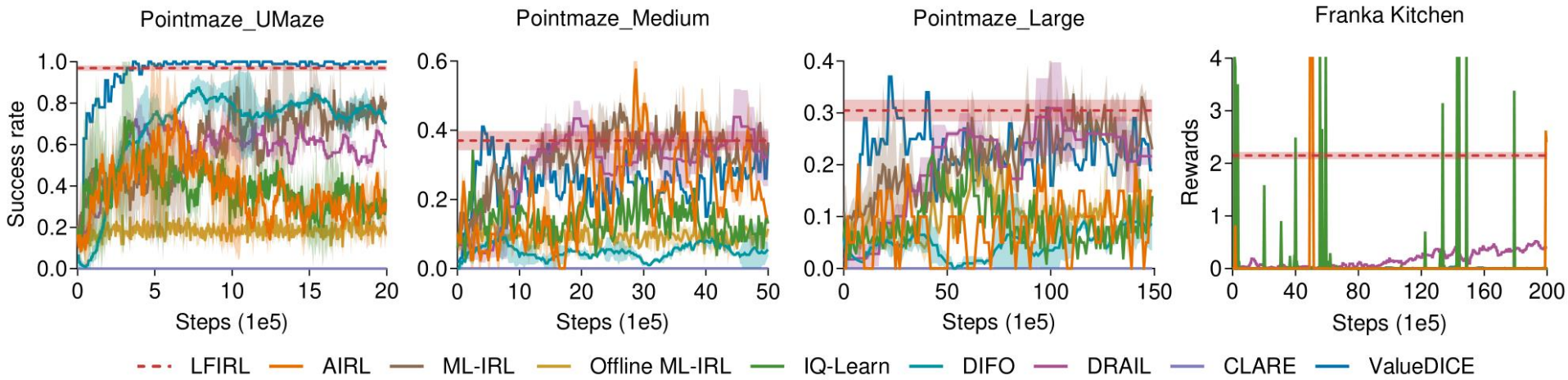
$$\mathcal{L}_r(\theta) = \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \mathcal{D}_E \cup \mathcal{D}_S} \left[\left(\boxed{\tilde{r}_{\theta}} - \boxed{\tilde{r}} \right)^2 \right] \tag{15}$$



Prediction **Fixed target**

Reward Recovery Quality

Downstream control · Figure 4



Trajectory-quality classification (%) · Table 1

Task	LFIRL	AIRL	IQ-Learn	DIFO	DRAIL	ML-IRL	Off. ML-IRL	CLARE	ValueDICE
Push-T	**74.17±1.96**	59.67±2.08	39.67±0.58	79.00±1.00	49.67±1.15	71.33±2.33	42.67±1.20	71.00±3.67	78.67±2.52
Pen	**97.17±2.75**	90.00±2.45	74.33±7.59	91.67±2.05	98.33±0.94	71.00±4.32	66.00±12.43	94.67±0.58	95.67±1.15

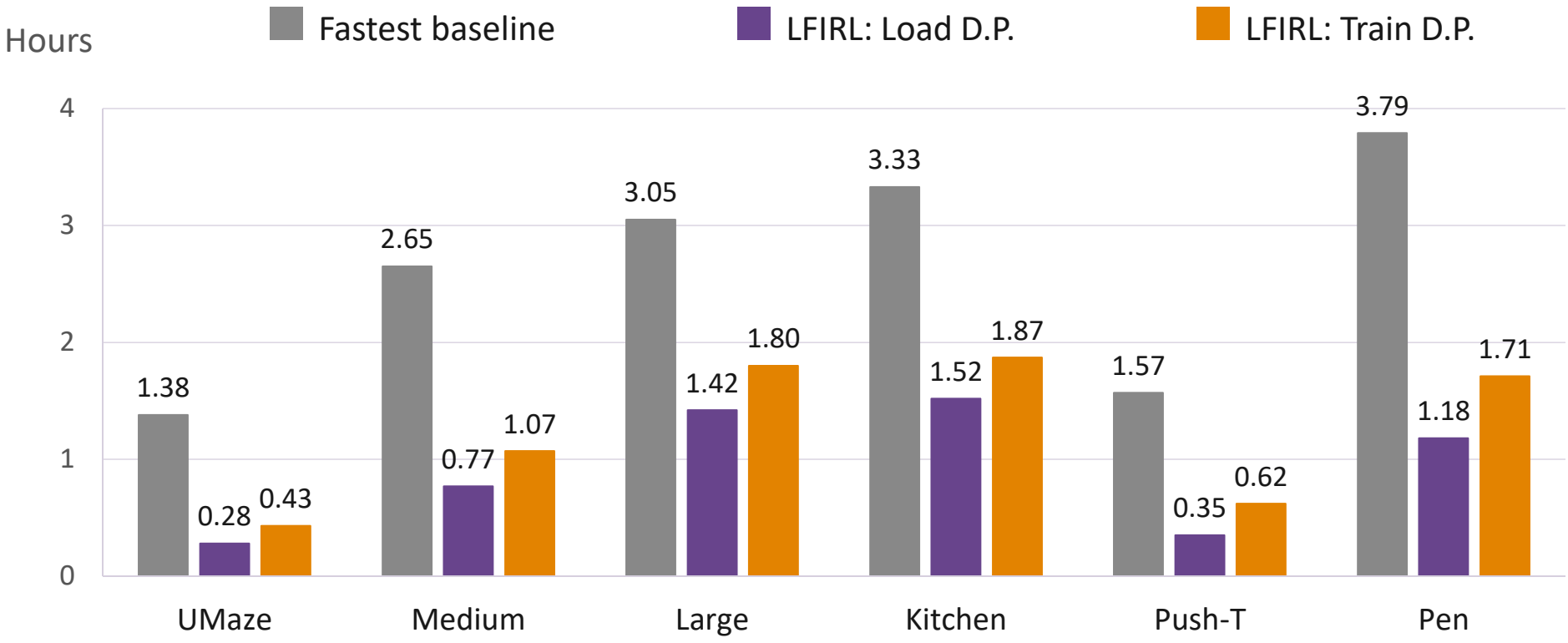
Ground-truth reward correlation · Table 3 (PCC / SCC)

Method	UMaze	Medium	Large	Kitchen	Push-T	Pen	Average
LFIRL	**0.83 / 0.91**	**0.76 / 0.82**	**0.84 / 0.87**	**0.67 / 0.81**	**0.72 / 0.81**	**0.79 / 0.85**	**0.77 / 0.85**
AIRL	0.65 / 0.86	0.63 / 0.74	0.77 / 0.75	0.44 / 0.63	0.69 / 0.74	0.75 / 0.71	0.66 / 0.74
IQ-Learn	0.62 / 0.74	0.59 / 0.62	0.65 / 0.70	0.34 / 0.51	0.48 / 0.65	0.47 / 0.62	0.53 / 0.64
DIFO	0.69 / 0.79	0.53 / 0.67	0.68 / 0.81	0.61 / 0.77	0.83 / 0.79	0.71 / 0.72	0.68 / 0.76
DRAIL	0.67 / 0.71	0.74 / 0.84	0.69 / 0.78	0.56 / 0.76	0.51 / 0.56	0.63 / 0.62	0.63 / 0.71
ML-IRL	0.78 / 0.75	0.67 / 0.72	0.64 / 0.71	0.63 / 0.69	0.66 / 0.73	0.78 / 0.69	0.69 / 0.72
Off. ML-IRL	0.49 / 0.52	0.47 / 0.58	0.42 / 0.62	0.45 / 0.54	0.45 / 0.53	0.53 / 0.43	0.47 / 0.54
CLARE	0.72 / 0.80	0.68 / 0.74	0.49 / 0.61	0.24 / 0.46	0.51 / 0.79	0.58 / 0.55	0.54 / 0.66
ValueDICE	0.72 / 0.81	0.61 / 0.79	0.74 / 0.71	0.53 / 0.56	0.65 / 0.76	0.57 / 0.63	0.64 / 0.71

Competitive reward usefulness; highest average PCC / SCC. Five runs; Table 1 reports mean ± SD.

Training Time

Average training time under the same budget

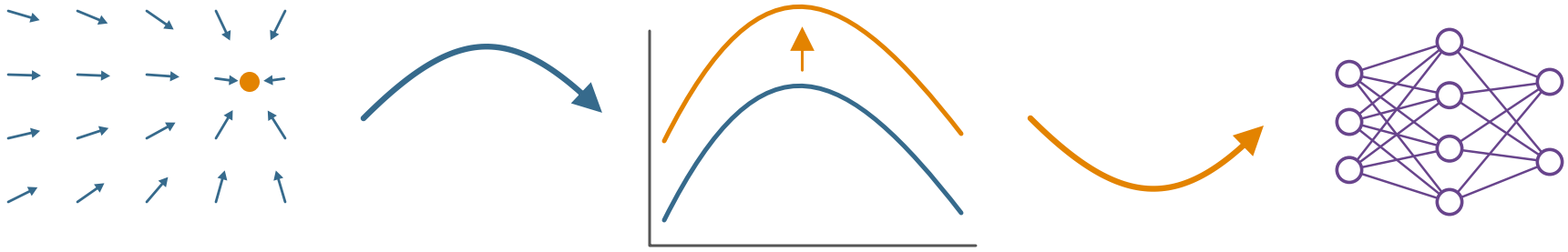


	UMaze	Medium	Large	Kitchen	Push-T	Pen
Baseline	AIRL	DRAIL	DRAIL	DRAIL	AIRL	CLARE
Load speedup	**4.9×**	**3.4×**	**2.1×**	**2.2×**	**4.5×**	**3.2×**
Train speedup	**3.2×**	**2.5×**	**1.7×**	**1.8×**	**2.5×**	**2.2×**

Source: Table 2. Speedup = fastest-baseline time / LFIRL time; Train D.P. includes policy pretraining.

Summary

From expert action directions to a reusable reward



Score supervision **Calibrated soft values** **Recovered reward**

- Three sequential stages, trained fully offline, no alternating reward and policy updates.
- Strong reward agreement across tasks. 2–3× faster training in most settings.

Thank you for your watching!