Trust Region Reward Optimization and Proximal Inverse Reward Optimization Algorithm

Yang Chen

NeurIPS 2025

NEURAL INFORMATION PROCESSING SYSTEMS

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

Reinforcement Learning

Philosophy & Logic

Human preferences exist

Aristotle, Plato, Epicurus...

Early

1662

Antoine Arnauld, Pierre Nicole

Port Royal Logic

Quantification of preferences

1823

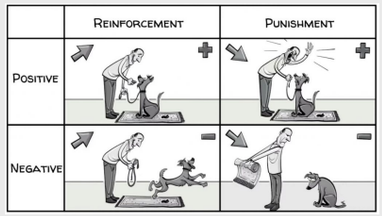
Jeremy Bentham

Utility Calculus

Math. Models & Theories

Increasing rewards

Operant Conditioning



1938

Pairwise preferences

Bradley-Terry Model

$$\begin{aligned}&\operatorname{logit}(\operatorname{Prob}(i \text{ beats } j))\\ &= \log\left(\frac{\operatorname{Prob}(i \text{ beats } j)}{\operatorname{Prob}(j \text{ beats } i)}\right)\\ &= \log\left(\frac{p_i}{p_j}\right) = \beta_i - \beta_j.\end{aligned}$$

1947

Von Neumann, Morgenstern

Utility Theorem

Time and context-independent preferences

1952

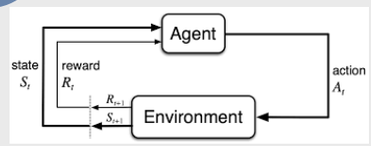
1957

Bellman **Equation**

Dynamic Programming Equation

Optimal policies exist

1960



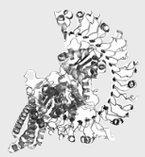
Markov Decision Process

Beat Human

Deep reinforcement learning

Alpha Fold 1

DeepMind



2018

PPO Algorithm

OpenAI

$$L^{CLIP}(\theta) = \hat{E}_t[min(r_t(\theta))\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)]$$

2017

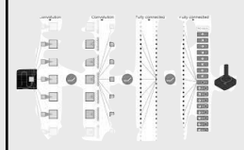
Alpha Go

DeepMind

2016

Deep Q learning

DeepMind



2013

LLMs

2025

deepseek

DRPO

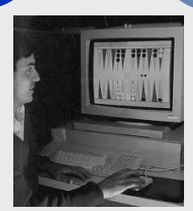
2022

ChatGPT

RLHF

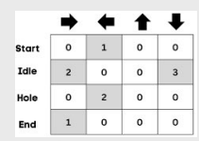
Reinforcement learning from human preferences from data

1995



IBM **TD Gammon**

1992



Q learning

1988

$$V(s) = V(s) + \alpha(r + \gamma V(s') - V(s))$$

Learning rate TD error

Temporal difference learning

Computing Optimal policies with approximation methods

Algorithms

Reinforcement Learning

Philosophy & Logic

Human preferences exist

Aristotle, Plato, Epicurus...

Early

1662

Antoine Arnauld, Pierre Nicole

Port Royal Logic

1823

Jeremy Bentham

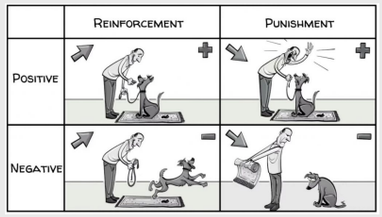
Utility Calculus

Quantification of preferences

Math. Models & Theories

Increasing rewards

Operant Conditioning



1938

1947

Von Neumann, Morgenstern

Utility Theorem

Time and context-independent preferences

Pairwise preferences

Bradley-Terry Model

$$\begin{aligned}&\operatorname{logit}\left(\operatorname{Prob}(i \text{ beats } j)\right)\\&= \log\left(\frac{\operatorname{Prob}(i \text{ beats } j)}{\operatorname{Prob}(j \text{ beats } i)}\right)\\&= \log\left(\frac{p_i}{p_j}\right) = \beta_i - \beta_j.\end{aligned}$$

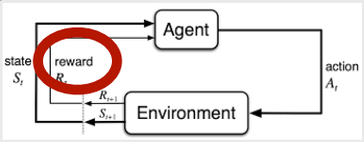
1952

1957

Bellman Equation

Dynamic Programming Equation

1960



Markov Decision Process

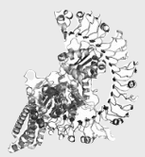
Optimal policies exist

Beat Human

Deep reinforcement learning

Alpha Fold 1

DeepMind



2018

PPO Algorithm

OpenAI

$$L^{CLIP}(\theta) = \hat{E}_t[\min(r_t(\theta))\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)]$$

2017

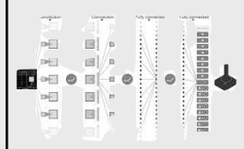
Alpha Go

DeepMind

2016

Deep Q learning

DeepMind



2013

LLMs

2025

DRPO

2022

RLHF

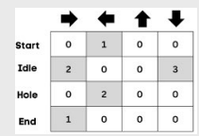
Reinforcement learning from human preferences from data

Reward Function !

1995

TD Gammon

1992



Q learning

1988

$$V(s) = V(s) + \alpha(r + \gamma V(s') - V(s))$$

Learning rate

TD error

Temporal difference learning

Computing Optimal policies with approximation methods

Algorithms

Imitation Learning and Inverse Reinforcement Learning

Human preferences exist

Aristotle, Plato, Epicurus...

Early

1662

Antoine Arnauld, Pierre Nicole

Port Royal Logic

Quantification of preferences

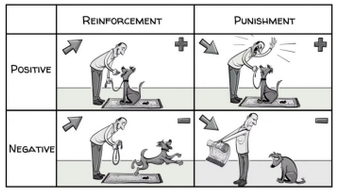
1823

Jeremy Bentham

Utility Calculus

Increasing rewards

Operant Conditioning



1938

Pairwise preferences

Bradley-Terry Model

$$\begin{aligned}&\operatorname{logit}\left(\operatorname{Prob}(i \text{ beats } j)\right)\\&= \log\left(\frac{\operatorname{Prob}(i \text{ beats } j)}{\operatorname{Prob}(j \text{ beats } i)}\right)\\&= \log\left(\frac{p_i}{p_j}\right) = \beta_i - \beta_j.\end{aligned}$$

1947

Von Neumann, Morgenstern

Utility Theorem

Time and context-independent preferences

1952

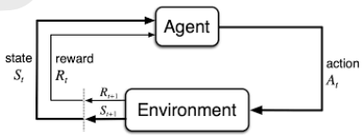
1957

Bellman Equation

Dynamic Programming Equation

Optimal policies exist

1960



Markov Decision Process

1988

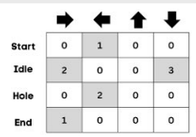
$V(s) = V(s) + \alpha(r + \gamma V(s') - V(s))$

Learning rate

TD error

Temporal difference learning

1992



Q learning

1995

IBM TD Gammon

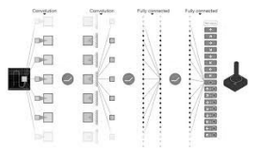
Computing Optimal policies with approximation methods

2013

Deep reinforcement learning

Deep Q learning

DeepMind



2016

Alpha Go

DeepMind

2017

PPO Algorithm

OpenAI

$$L^{CLIP}(\theta) = \hat{E}_t[min(r_t(\theta))\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)]$$

2018

Alpha Fold 1

DeepMind

2022

ChatGPT

ChatGPT

2025

DeepSeek

Reinforcement learning from human preferences

Imitation Learning and Inverse Reinforcement Learning

Deep Learning, Complex Tasks

Deep reinforcement learning

Multiple agents

Classic Machine Learning, Simple Tasks

Reinforcement learning from human preferences

Computing Optimal policies with approximation methods

2019 — 2017 — 2016 — 2008 — 2007 — 2004 — 2000

2025 — 2022 — 2018 — 2017 — 2016 — 2013 — 1995 — 1992 — 1988

Multi-agent IRL

Adversarial IRL

Generative adversarial imitation learning

Maximum Entropy IRL

Guided Cost Learning

Bayesian IRL

Apprenticeship learning

Concept of IRL

Alpha Fold 1

PPO Algorithm

Alpha Go

Deep Q learning

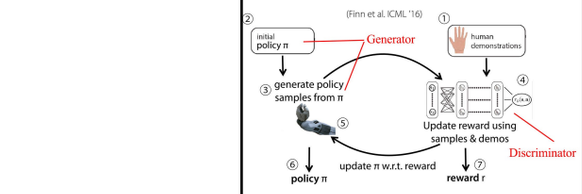
DeepSeek

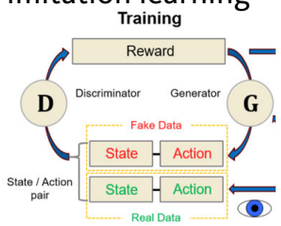
ChatGPT

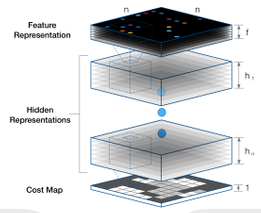
TD Gammon

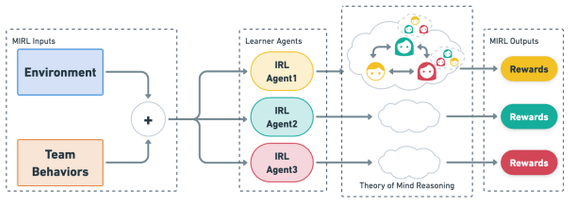
Q learning

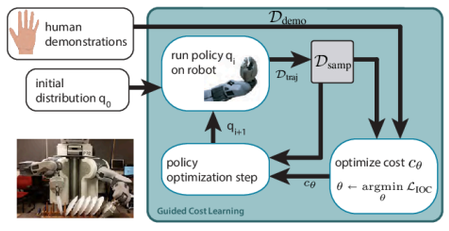
Temporal difference learning

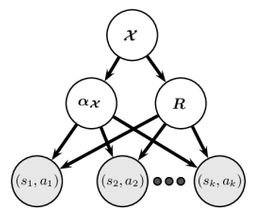


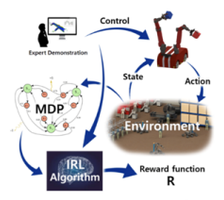


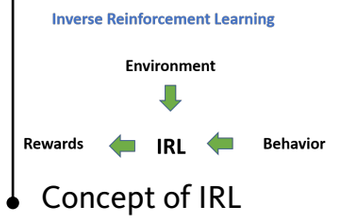


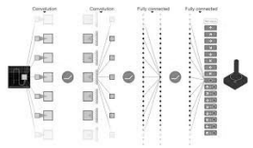


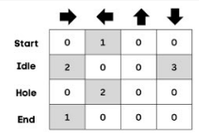


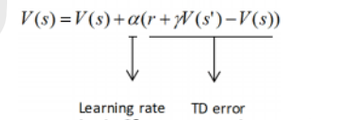








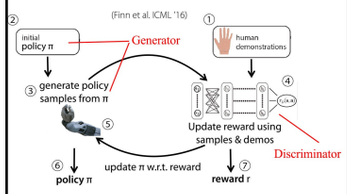




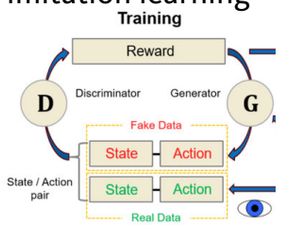
Imitation Learning and Inverse Reinforcement Learning

Deep Learning, Complex Tasks

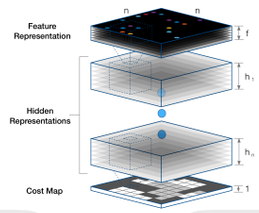
Adversarial IRL



Generative adversarial imitation learning

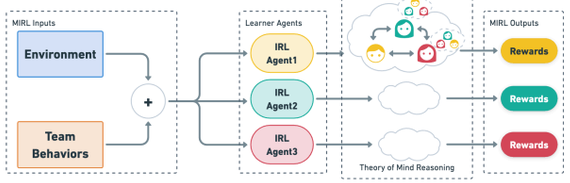


Maximum Entropy IRL



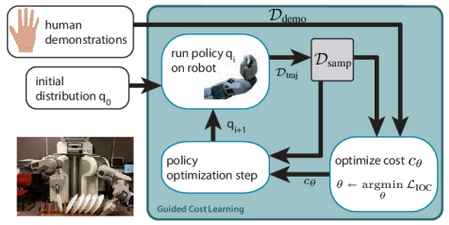
2019 — 2017 — 2016 — 2008 — 2007 — 2004 — 2000

Multi-agent IRL

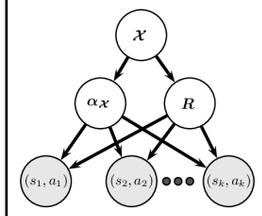


Multiple agents

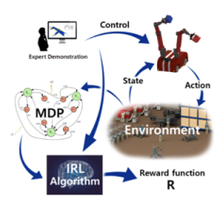
Guided Cost Learning



Bayesian IRL



Apprenticeship learning



Inverse Reinforcement Learning

Concept of IRL

Classic Machine Learning, Simple Tasks

2025 — 2022

deepseek — DeepSeek

ChatGPT — ChatGPT

Reinforcement learning from human preferences

Deep reinforcement learning

Alpha Fold 1

DeepMind

PPO Algorithm

OpenAI

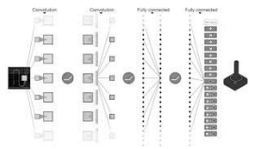
$$L^{CLIP}(\theta) = \hat{E}_t[min(r_t(\theta))\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)]$$

Alpha Go

DeepMind

Deep Q learning

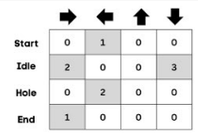
DeepMind



2018 — 2017 — 2016 — 2013

1995 — 1992 — 1988

TD Gammon



Q learning

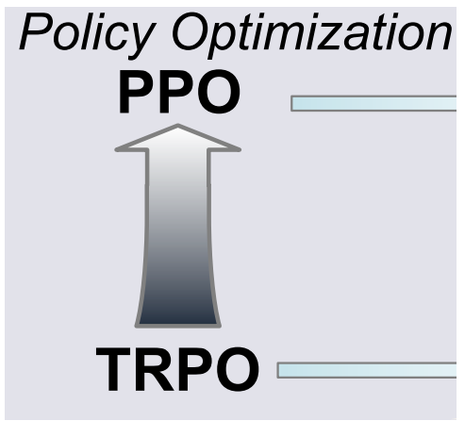
$$V(s) = V(s) + \alpha(r + \gamma V(s') - V(s))$$

Learning rate — TD error

Temporal difference learning

Computing Optimal policies with approximation methods

The Success of Stable Reinforcement Learning



Sources of pictures: *https://avandekleut.github.io/assets/ppo/trust-region.png* *https://huggingface.co/blog/assets/93_deep_rl_ppo/cliff.jpg*

The Success of Stable Reinforcement Learning

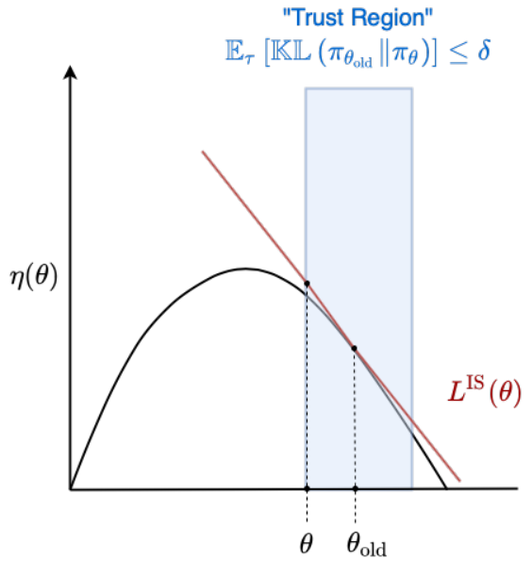
Policy Optimization

PPO

TRPO

$$\underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right]$$

$$\text{subject to} \quad \hat{\mathbb{E}}_t[\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \leq \delta.$$



Sources of pictures: https://avandekleut.github.io/assets/ppo/trust-region.png https://huggingface.co/blog/assets/93_deep_rl_ppo/cliff.jpg

The Success of Stable Reinforcement Learning

PPO's Clipped surrogate objective function

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

The ratio function

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$

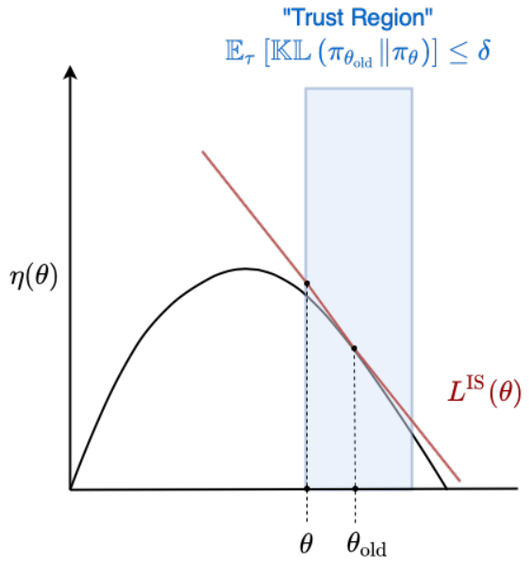
Policy Optimization

PPO

TRPO

$$\underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right]$$

$$\text{subject to} \quad \hat{\mathbb{E}}_t[\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \leq \delta.$$



Sources of pictures: https://avandekleut.github.io/assets/ppo/trust-region.png https://huggingface.co/blog/assets/93_deep_rl_ppo/cliff.jpg

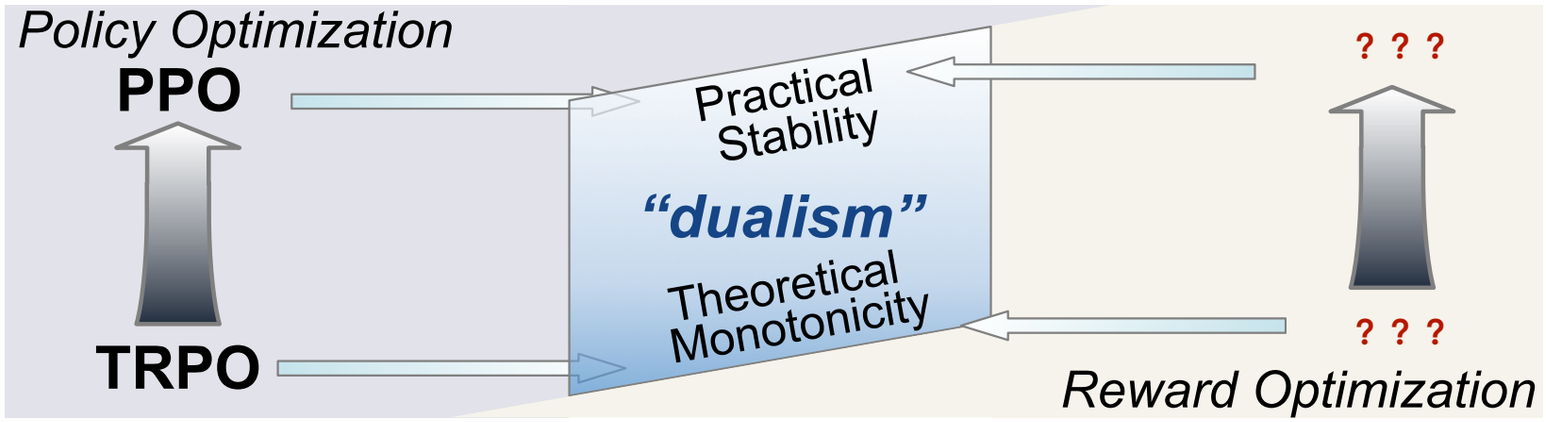
The Success of Stable Reinforcement Learning

PPO's Clipped surrogate objective function

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

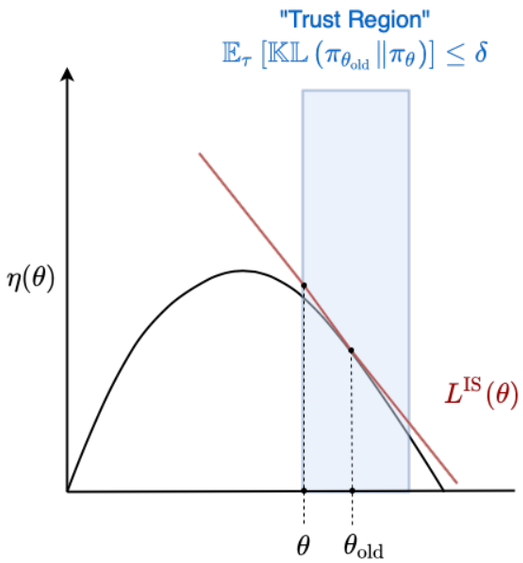
The ratio function

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$



$$\underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right]$$

$$\text{subject to} \quad \hat{\mathbb{E}}_t[\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \leq \delta.$$



Sources of pictures: https://avandekleut.github.io/assets/ppo/trust-region.png https://huggingface.co/blog/assets/93_deep_rl_ppo/cliff.jpg

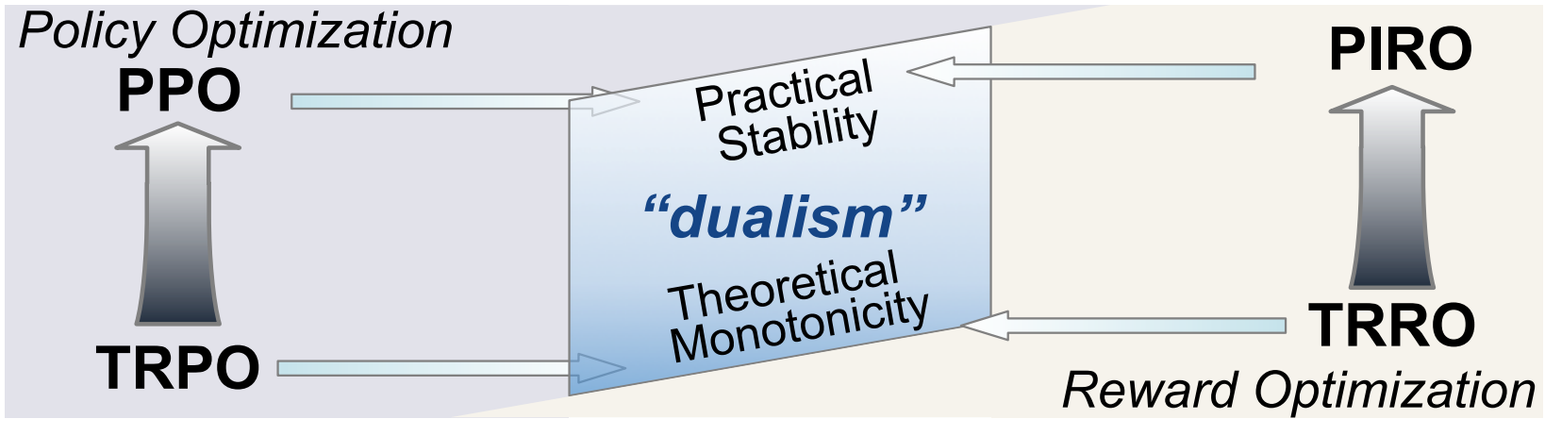
The Success of Stable Reinforcement Learning

PPO's Clipped surrogate objective function

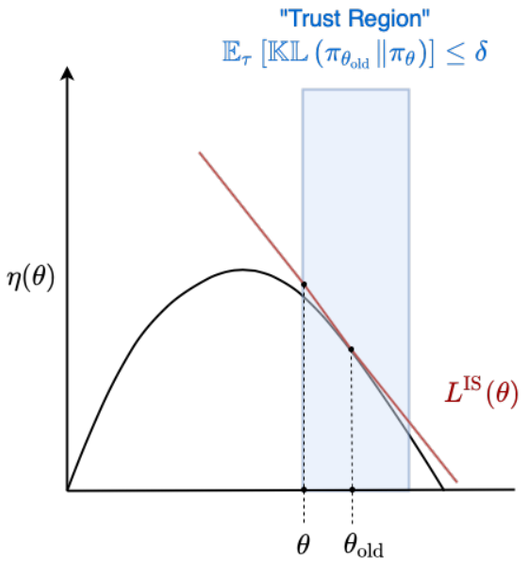
$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

The ratio function

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$



$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] \\ \text{subject to} \quad & \hat{\mathbb{E}}_t[\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \leq \delta. \end{aligned}$$



Sources of pictures: https://avandekleut.github.io/assets/ppo/trust-region.png https://huggingface.co/blog/assets/93_deep_rl_ppo/cliff.jpg

The Success of Stable Reinforcement Learning

PPO's Clipped surrogate objective function

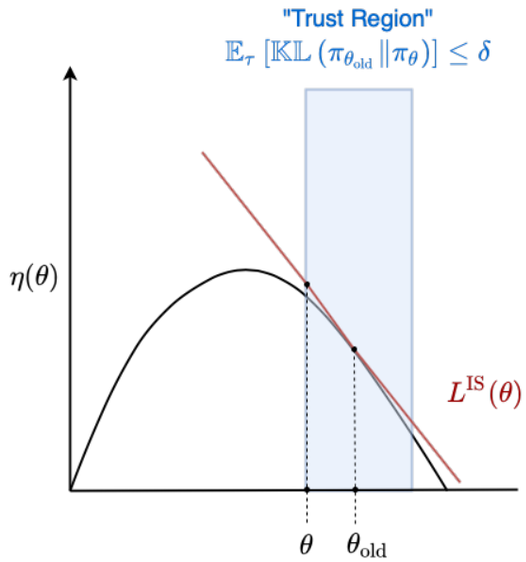
$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

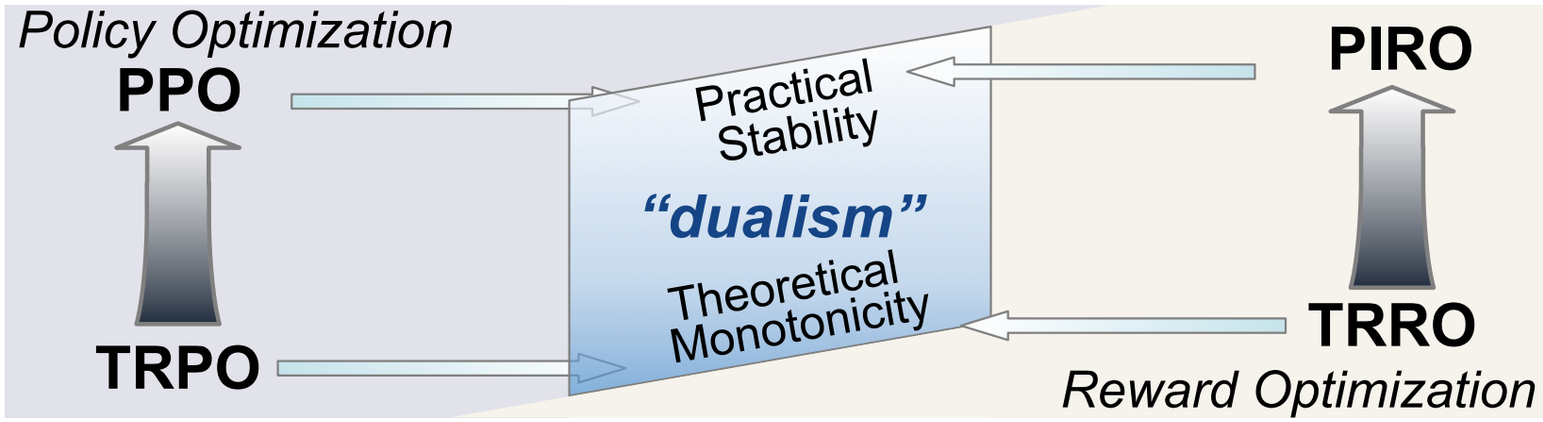
The ratio function

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$

$$\underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right]$$

$$\text{subject to} \quad \hat{\mathbb{E}}_t[\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \leq \delta.$$





Proximal Inverse Reward Optimization

Trust Region Reward Optimization

Sources of pictures: https://avandekleut.github.io/assets/ppo/trust-region.png https://huggingface.co/blog/assets/93_deep_rl_ppo/cliff.jpg

Concepts of Inverse Reinforcement Lerarning

Elements of Inverse Reinforcement Learning

$$\min_{\pi \in \Pi} \max_{r \in \mathcal{R}} J(\pi_E, r) - J(\pi, r) = \mathbb{E}_{\rho^{\pi_E}}[r(\mathbf{s}, \mathbf{a})] - (\mathbb{E}_{\rho^{\pi}}[r(\mathbf{s}, \mathbf{a})] + \mathcal{H}(\pi)). \quad \text{(MaxEnt-IRL)}$$

- π_E: expert policy
- r: reward function
- ρ^{π}: occupancy measure (state-action visitation frequency)
- $\mathcal{H}(\pi)$: policy entropy

References:
[MaxEntIRL] *Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008, July). Maximum entropy inverse reinforcement learning. In AAAI (Vol. 8, pp. 1433-1438).*
[ML-IRL] *Zeng, S., Li, C., Garcia, A., & Hong, M. (2022). Maximum-likelihood inverse reinforcement learning with finite-time guarantees. Advances in Neural Information Processing Systems, 35, 10122-10135.*

Concepts of Inverse Reinforcement Lerarning

Elements of Inverse Reinforcement Learning

$$\min_{\pi\in\Pi}\max_{r\in\mathcal{R}} J(\pi_E, r) - J(\pi, r) = \mathbb{E}_{\rho^{\pi_E}}[r(\mathbf{s},\mathbf{a})] - (\mathbb{E}_{\rho^{\pi}}[r(\mathbf{s},\mathbf{a})] + \mathcal{H}(\pi)). \qquad \text{(MaxEnt-IRL)}$$

- π_E: expert policy
- r: reward function
- ρ^{π}: occupancy measure (state-action visitation frequency)
- $\mathcal{H}(\pi)$: policy entropy

The Likelihood Objective of Inverse Reinforcement Learning

$$\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) := \mathbb{E}_{\rho^{\pi_E}}[\log \pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{s})] \quad s.t. \quad \pi_{\boldsymbol{\theta}} = \arg\max_{\pi} \mathbb{E}_{\pi}[r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})] \qquad \text{(ML-IRL)}$$

References:

[MaxEntIRL] *Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008, July). Maximum entropy inverse reinforcement learning. In AAAI (Vol. 8, pp. 1433-1438).*

[ML-IRL] *Zeng, S., Li, C., Garcia, A., & Hong, M. (2022). Maximum-likelihood inverse reinforcement learning with finite-time guarantees. Advances in Neural Information Processing Systems, 35, 10122-10135.*

Concepts of Inverse Reinforcement Lerarning

Elements of Inverse Reinforcement Learning

$$\min_{\pi\in\Pi}\max_{r\in\mathcal{R}} J(\pi_E, r) - J(\pi, r) = \mathbb{E}_{\rho^{\pi_E}}[r(\mathbf{s},\mathbf{a})] - (\mathbb{E}_{\rho^{\pi}}[r(\mathbf{s},\mathbf{a})] + \mathcal{H}(\pi)). \qquad \text{(MaxEnt-IRL)}$$

- π_E: expert policy
- r: reward function
- ρ^{π}: occupancy measure (state-action visitation frequency)
- $\mathcal{H}(\pi)$: policy entropy

The Likelihood Objective of Inverse Reinforcement Learning

$$\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) := \mathbb{E}_{\rho^{\pi_E}}[\log \pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{s})] \quad s.t. \quad \pi_{\boldsymbol{\theta}} = \arg\max_{\pi} \mathbb{E}_{\pi}[r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})] \qquad \text{(ML-IRL)}$$

The Equivalence Between Two Objectives

$$\ell(\boldsymbol{\theta}) = \mathbb{E}_{\rho^{\pi_E}}[r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\mathbf{s}_0\sim\eta}[V_{r_{\boldsymbol{\theta}}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_0)] = J(\pi_E, r_{\boldsymbol{\theta}}) - J(\pi_{\boldsymbol{\theta}}, r_{\boldsymbol{\theta}}),$$

$$\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}) = \mathbb{E}_{\rho^{\pi_E}}[\nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\rho^{\pi_{\boldsymbol{\theta}}}}[\nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})].$$

References:
[MaxEntIRL] *Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008, July). Maximum entropy inverse reinforcement learning. In AAAI (Vol. 8, pp. 1433-1438).*
[ML-IRL] *Zeng, S., Li, C., Garcia, A., & Hong, M. (2022). Maximum-likelihood inverse reinforcement learning with finite-time guarantees. Advances in Neural Information Processing Systems, 35, 10122-10135.*

Concepts of Inverse Reinforcement Lerarning

Elements of Inverse Reinforcement Learning

$$\min_{\pi\in\Pi}\max_{r\in\mathcal{R}} J(\pi_E, r) - J(\pi, r) = \mathbb{E}_{\rho^{\pi_E}}[r(\mathbf{s},\mathbf{a})] - (\mathbb{E}_{\rho^{\pi}}[r(\mathbf{s},\mathbf{a})] + \mathcal{H}(\pi)). \qquad \text{(MaxEnt-IRL)}$$

- π_E: expert policy
- r: reward function
- ρ^{π}: occupancy measure (state-action visitation frequency)
- $\mathcal{H}(\pi)$: policy entropy

The Likelihood Objective of Inverse Reinforcement Learning

$$\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) := \mathbb{E}_{\rho^{\pi_E}}[\log \pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{s})] \quad s.t. \quad \pi_{\boldsymbol{\theta}} = \arg\max_{\pi} \mathbb{E}_{\pi}[r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})] \qquad \text{(ML-IRL)}$$

The Equivalence Between Two Objectives

$$\ell(\boldsymbol{\theta}) = \mathbb{E}_{\rho^{\pi_E}}[r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\mathbf{s}_0\sim\eta}[V_{r_{\boldsymbol{\theta}}}^{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_0)] = J(\pi_E, r_{\boldsymbol{\theta}}) - J(\pi_{\boldsymbol{\theta}}, r_{\boldsymbol{\theta}}),$$

$$\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}) = \mathbb{E}_{\rho^{\pi_E}}[\nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\rho^{\pi_{\boldsymbol{\theta}}}}[\nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})].$$

Insight

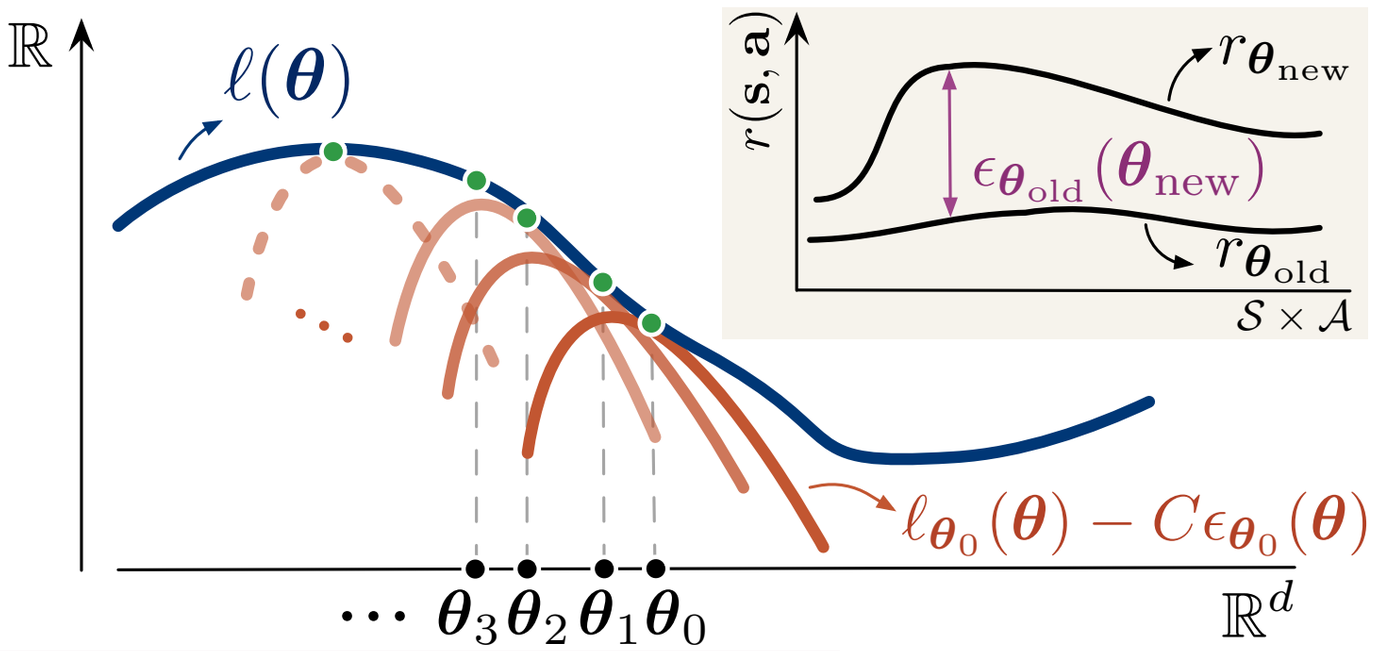
The stability of IRL can be achieved by monotonically improving the likelihood of expert demonstrations, which is equivalent to reducing the return gap between the expert policy and the reward-induced policy.

References:
[MaxEntIRL] *Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008, July). Maximum entropy inverse reinforcement learning. In AAAI (Vol. 8, pp. 1433-1438).*
[ML-IRL] *Zeng, S., Li, C., Garcia, A., & Hong, M. (2022). Maximum-likelihood inverse reinforcement learning with finite-time guarantees. Advances in Neural Information Processing Systems, 35, 10122-10135.*

Trust Region Reward Optimization (TRRO)

Local Approximation of the Likelihood Objective

$$\underbrace{\ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}_{\text{old}}) = \ell(\boldsymbol{\theta}_{\text{old}})}_{\equiv \mathbb{E}_{\rho^{\pi_E}}[r_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\mathbf{s}_0 \sim \eta}[V^{\pi_{\text{old}}}_{r_{\boldsymbol{\theta}_{\text{old}}}}(\mathbf{s}_0)]} \quad \textit{and} \quad \underbrace{\nabla_{\boldsymbol{\theta}} \ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{old}}} = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{old}}}}_{\equiv \mathbb{E}_{\rho^{\pi_E}}[\nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\rho^{\pi_{\text{old}}}}[\nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})]|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{old}}}} .$$



Trust Region Reward Optimization (TRRO)

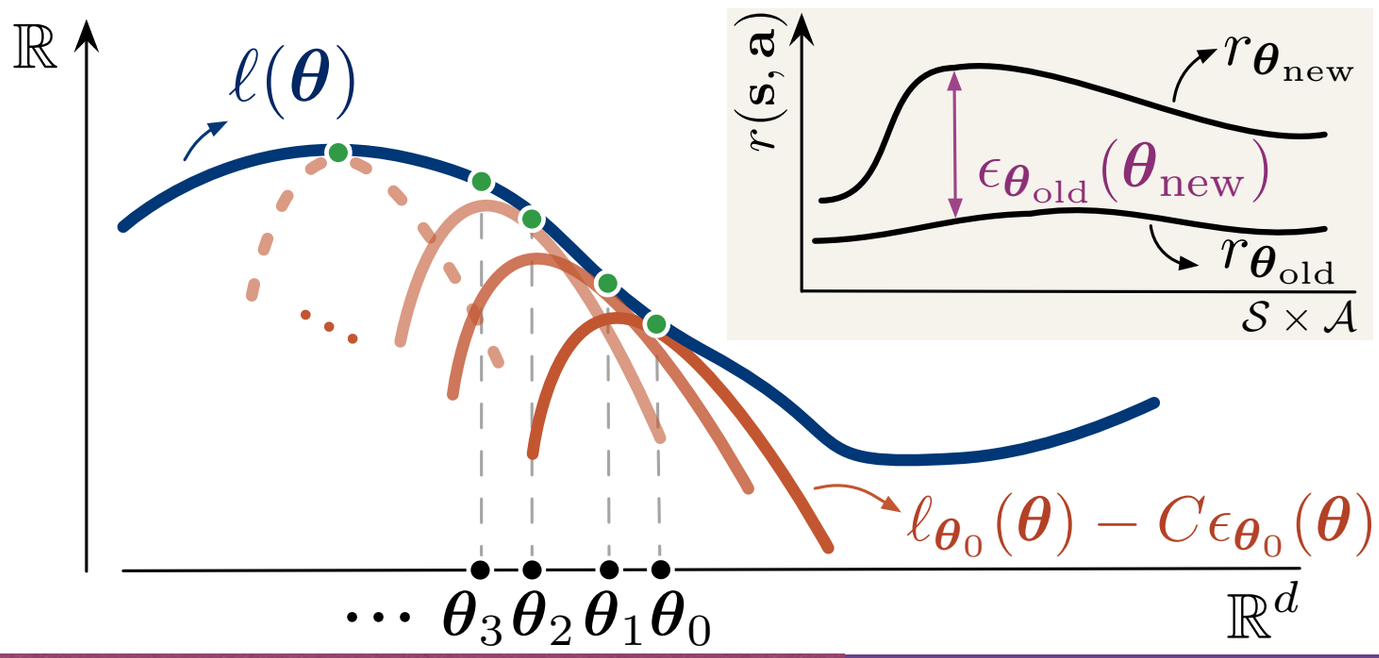
Local Approximation of the Likelihood Objective

$$\underbrace{\ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}_{\text{old}}) = \ell(\boldsymbol{\theta}_{\text{old}})}_{\equiv \mathbb{E}_{\rho^{\pi_E}}[r_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\mathbf{s}_0\sim\eta}[V^{\pi_{\text{old}}}_{r_{\boldsymbol{\theta}_{\text{old}}}}(\mathbf{s}_0)]} \quad \textit{and} \quad \underbrace{\nabla_{\boldsymbol{\theta}}\ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{old}}} = \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{old}}}}_{\equiv \mathbb{E}_{\rho^{\pi_E}}[\nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\rho^{\pi_{\text{old}}}}[\nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})]|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{old}}}} .$$

The Guarantee of Monotonic Improvement of the Likelihood

$$\ell(\boldsymbol{\theta}_{\text{new}}) \geq \ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}_{\text{new}}) - C\epsilon_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}_{\text{new}}), \textit{ where the constant}$$

$$C = \frac{2|\mathcal{A}|}{(1-\gamma)^2} + \frac{(5-\gamma)|\mathcal{A}|R + (\gamma - \gamma^2 + 2)|\mathcal{A}|\log|\mathcal{A}|}{(1-\gamma)^4}.$$



Trust Region Reward Optimization (TRRO)

Local Approximation of the Likelihood Objective

$$\underbrace{\ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}_{\text{old}}) = \ell(\boldsymbol{\theta}_{\text{old}})}_{\equiv \mathbb{E}_{\rho^{\pi_E}}[r_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\mathbf{s}_0\sim\eta}[V^{\pi_{\text{old}}}_{r_{\boldsymbol{\theta}_{\text{old}}}}(\mathbf{s}_0)]} \quad \textit{and} \quad \underbrace{\nabla_{\boldsymbol{\theta}}\ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{old}}} = \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{old}}}}_{\equiv \mathbb{E}_{\rho^{\pi_E}}[\nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\rho^{\pi_{\text{old}}}}[\nabla_{\boldsymbol{\theta}} r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a})]|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{old}}}}.$$

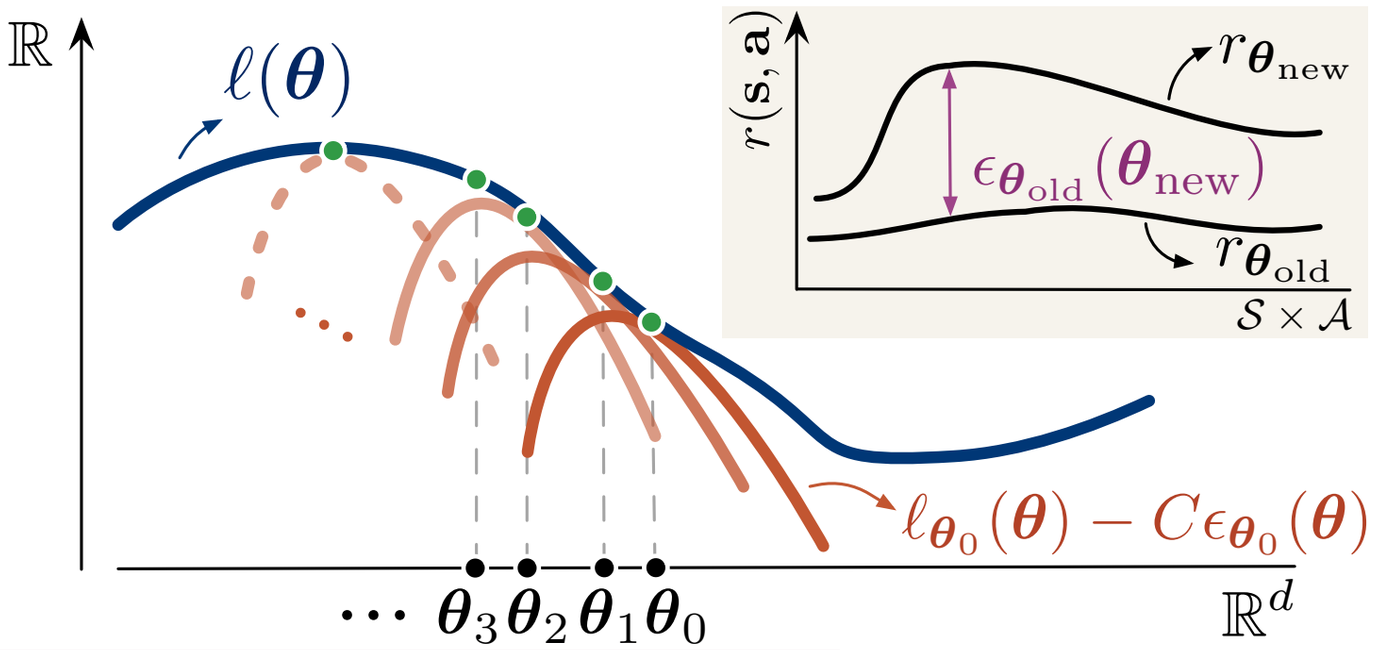
The Guarantee of Monotonic Improvement of the Likelihood

$$\ell(\boldsymbol{\theta}_{\text{new}}) \geq \ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}_{\text{new}}) - C\epsilon_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}_{\text{new}}), \textit{ where the constant}$$

$$C = \frac{2|\mathcal{A}|}{(1-\gamma)^2} + \frac{(5-\gamma)|\mathcal{A}|R + (\gamma - \gamma^2 + 2)|\mathcal{A}|\log|\mathcal{A}|}{(1-\gamma)^4}.$$

The Procedure of TRRO

$$\pi = \arg\max_{\pi} J(\pi, r_{\boldsymbol{\theta}_{\text{old}}}) \overset{\substack{\pi\to\pi_{\text{old}} \\ \rightleftharpoons \\ \boldsymbol{\theta}_{\text{old}}\leftarrow\boldsymbol{\theta}_{\text{new}}}}{} \boldsymbol{\theta}_{\text{new}} = \arg\max_{\boldsymbol{\theta}} \ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) - C\epsilon_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}). \qquad \text{(TRRO)}$$



Proximal Inverse Reward Optimization Algorithm (PIRO)

Adaptive Reward Update

$$\bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) := \left(\sum\nolimits_{(\mathbf{s},\mathbf{a}) \in \hat{\mathcal{D}}_E \cup \mathcal{D}_S} \left(r_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s},\mathbf{a}) - r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) \right)^2 \right)^{1/2}$$

$$\boldsymbol{\theta}_{\text{new}} = \arg\max_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) := \ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) - \mu \bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}).$$

$$\text{If } \bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) > \bar{\epsilon}^{\text{target}} \times x, \text{ then } \mu \leftarrow \mu \times y; \quad \text{If } \bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) < \bar{\epsilon}^{\text{target}} / x, \text{ then } \mu \leftarrow \mu / y,$$

Proximal Inverse Reward Optimization Algorithm (PIRO)

Adaptive Reward Update

$$\bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) := \left(\sum\nolimits_{(\mathbf{s},\mathbf{a}) \in \hat{\mathcal{D}}_E \cup \mathcal{D}_S} \left(r_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s},\mathbf{a}) - r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) \right)^2 \right)^{1/2}$$

$$\boldsymbol{\theta}_{\text{new}} = \arg\max_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) := \ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) - \mu \bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}).$$

$$\text{If } \bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) > \bar{\epsilon}^{\text{target}} \times x, \text{ then } \mu \leftarrow \mu \times y; \quad \text{If } \bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) < \bar{\epsilon}^{\text{target}} / x, \text{ then } \mu \leftarrow \mu / y,$$

Practical Policy Optimization

$$\pi \leftarrow k \text{ SAC rounds with } r_{\boldsymbol{\theta}_{\text{old}}}, \pi_{\text{old}}$$

Proximal Inverse Reward Optimization Algorithm (PIRO)

Adaptive Reward Update

$$\bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) := \left(\sum\nolimits_{(\mathbf{s},\mathbf{a}) \in \hat{\mathcal{D}}_E \cup \mathcal{D}_S} \left(r_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s},\mathbf{a}) - r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) \right)^2 \right)^{1/2}$$

$$\boldsymbol{\theta}_{\text{new}} = \arg\max_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) := \ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) - \mu \bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}).$$

$$\text{If } \bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) > \bar{\epsilon}^{\text{target}} \times x, \text{ then } \mu \leftarrow \mu \times y; \quad \text{If } \bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) < \bar{\epsilon}^{\text{target}} / x, \text{ then } \mu \leftarrow \mu / y,$$

Practical Policy Optimization

$$\pi \leftarrow k \text{ SAC rounds with } r_{\boldsymbol{\theta}_{\text{old}}}, \pi_{\text{old}}$$

PIRO Algorithm

$$\pi \leftarrow k \text{ SAC rounds with } r_{\boldsymbol{\theta}_{\text{old}}}, \pi_{\text{old}} \underset{\boldsymbol{\theta}_{\text{old}} \leftarrow \boldsymbol{\theta}_{\text{new}}}{\overset{\pi \rightarrow \pi_{\text{old}}}{\rightleftharpoons}} \boldsymbol{\theta}_{\text{new}} \leftarrow n \text{ grad. steps with } \nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}). \quad \text{(PIRO)}$$

Proximal Inverse Reward Optimization Algorithm (PIRO)

Adaptive Reward Update

$$\bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) := \left(\sum\nolimits_{(\mathbf{s},\mathbf{a}) \in \hat{\mathcal{D}}_E \cup \mathcal{D}_S} \left(r_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{s},\mathbf{a}) - r_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{a}) \right)^2 \right)^{1/2}$$

$$\boldsymbol{\theta}_{\text{new}} = \arg\max_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) := \ell_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) - \mu \bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}).$$

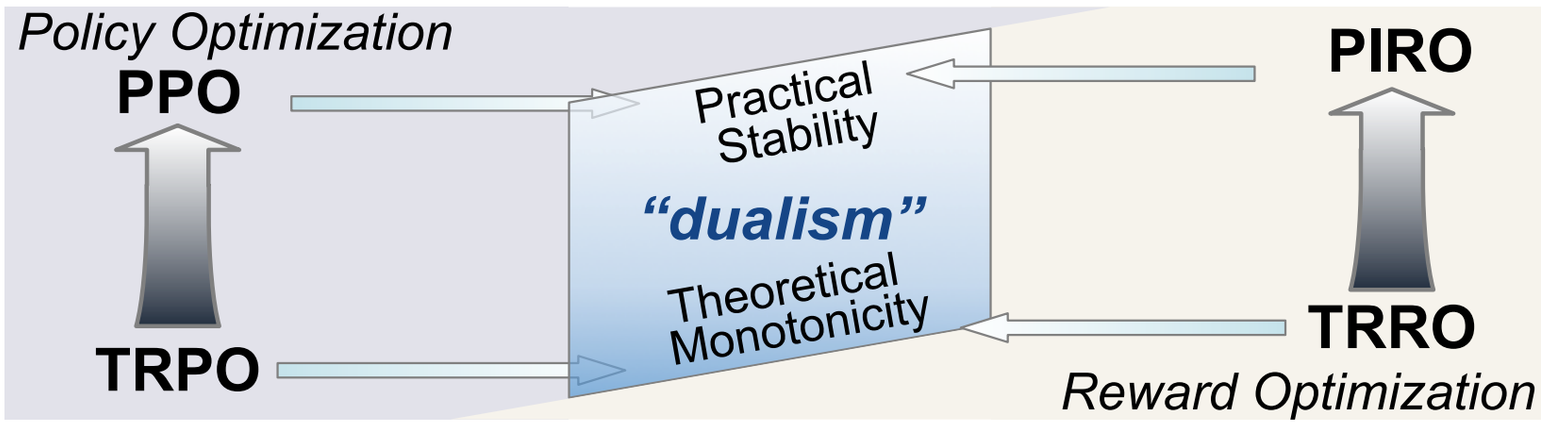
$$\text{If } \bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) > \bar{\epsilon}^{\text{target}} \times x, \text{ then } \mu \leftarrow \mu \times y; \quad \text{If } \bar{\epsilon}_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}) < \bar{\epsilon}^{\text{target}} / x, \text{ then } \mu \leftarrow \mu / y,$$

Practical Policy Optimization

$$\pi \leftarrow k \text{ SAC rounds with } r_{\boldsymbol{\theta}_{\text{old}}}, \pi_{\text{old}}$$

PIRO Algorithm

$$\pi \leftarrow k \text{ SAC rounds with } r_{\boldsymbol{\theta}_{\text{old}}}, \pi_{\text{old}} \underset{\boldsymbol{\theta}_{\text{old}} \leftarrow \boldsymbol{\theta}_{\text{new}}}{\overset{\pi \rightarrow \pi_{\text{old}}}{\rightleftharpoons}} \boldsymbol{\theta}_{\text{new}} \leftarrow n \text{ grad. steps with } \nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}_{\text{old}}}(\boldsymbol{\theta}). \quad \text{(PIRO)}$$



REWARD RECOVERY AND POLICY IMITATION

Table 1: **Averaged Rewards** (five independent runs) on five MuJoCo and four Gym Robotics tasks.

	Task	Expert	IL		Adv. IRL (Online)				Adv. (Offline)	Non-Adv. Online			Non-Adv. Offline		PIRO	*Gain*
			BC	GAIL	MM	AIRL	FILTER	HyPE	DAC	IQ	ML-IRL	f-IRL	CSIL	P^2IL		
MuJoCo	Ant-v4	5926.2	1631.5	996.9	-304.0	991.4	-376.3	2800.5	923.8	3589.8	5382.5	980.4	420.7	976.6	**5967.2**	+584.7
	Humanoid-v4	5501.0	418.1	508.4	367.2	281.4	291.7	717.5	76.3	1847.5	5573.4	470.4	–	–	**5954.9**	+381.5
	Walker2d-v4	5524.5	384.4	4158.1	70.4	72.8	77.7	1478.7	-3.0	3023.0	4794.7	243.8	686.1	1054.0	**5643.7**	+849.0
	Hopper-v4	3632.8	1034.4	**3535.7**	57.8	13.5	37.3	2593.7	3321.6	3424.5	3316.4	361.7	6.7	25.8	3362.0	-173.7
	Halfcheetah-v4	12266.1	221.2	1298.8	20.3	2251.4	0.3	6473.4	9645.0	3825.5	11873.2	-0.7	-107.2	-0.1	**12587.4**	+714.2
Robotics	AntMaze-UMazeDense-v4	35.6	8.8	5.2	5.1	4.5	6.1	11.9	–	3.9	4.2	3.6	–	3.4	**25.7**	+13.8
	AntMaze-MediumDense-v4	26.9	1.1	1.3	3.4	2.6	1.9	3.0	–	3.4	0.9	1.1	–	2.9	**9.4**	+6.0
	AntMaze-LargeDense-v4	11.5	1.1	0.9	1.7	3.4	0.6	1.5	–	0.8	0.3	0.9	–	0.2	**8.8**	+5.4
	AdroitHandePen-Human-v1	1062.5	44.1	-8.7	-344.3	-593.9	-685.4	-866.7	–	-751.9	-251.2	-65.3	–	-61.2	**254.0**	+209.9
	runtime per iteration	-	-	3-14s	8-79s	5-8s	9-41s	11-70s	135-142s	7-57s	93-166s	16-85s	68-90s	20-111s	96-178s	–

Note: DAC, CSIL and P^2IL are not evaluated on certain tasks due to compatibility issues cause by version conflicts. Specifically, the current implementations of DAC and P^2IL are incompatible with the current Gymnasium Robotics suite, while P^2IL and CSIL are incompatible with the Humanoid version used in testing other algorithms.

Experimental Evaluation

Reward Recovery and Policy Imitation

Table 1: **Averaged Rewards** (five independent runs) on five MuJoCo and four Gym Robotics tasks.

	Task	Expert	IL		Adv. IRL (Online)				Adv. (Offline)	Non-Adv. Online			Non-Adv. Offline		PIRO	*Gain*
			BC	GAIL	MM	AIRL	FILTER	HyPE	DAC	IQ	ML-IRL	f-IRL	CSIL	P^2IL		
MuJoCo	Ant-v4	5926.2	1631.5	996.9	-304.0	991.4	-376.3	2800.5	923.8	3589.8	5382.5	980.4	420.7	976.6	**5967.2**	+584.7
	Humanoid-v4	5501.0	418.1	508.4	367.2	281.4	291.7	717.5	76.3	1847.5	5573.4	470.4	–	–	**5954.9**	+381.5
	Walker2d-v4	5524.5	384.4	4158.1	70.4	72.8	77.7	1478.7	-3.0	3023.0	4794.7	243.8	686.1	1054.0	**5643.7**	+849.0
	Hopper-v4	3632.8	1034.4	**3535.7**	57.8	13.5	37.3	2593.7	3321.6	3424.5	3316.4	361.7	6.7	25.8	3362.0	-173.7
	Halfcheetah-v4	12266.1	221.2	1298.8	20.3	2251.4	0.3	6473.4	9645.0	3825.5	11873.2	-0.7	-107.2	-0.1	**12587.4**	+714.2
Robotics	AntMaze-UMazeDense-v4	35.6	8.8	5.2	5.1	4.5	6.1	11.9	–	3.9	4.2	3.6	–	3.4	**25.7**	+13.8
	AntMaze-MediumDense-v4	26.9	1.1	1.3	3.4	2.6	1.9	3.0	–	3.4	0.9	1.1	–	2.9	**9.4**	+6.0
	AntMaze-LargeDense-v4	11.5	1.1	0.9	1.7	3.4	0.6	1.5	–	0.8	0.3	0.9	–	0.2	**8.8**	+5.4
	AdroitHandePen-Human-v1	1062.5	44.1	-8.7	-344.3	-593.9	-685.4	-866.7	–	-751.9	-251.2	-65.3	–	-61.2	**254.0**	+209.9
	runtime per iteration	-	-	3-14s	8-79s	5-8s	9-41s	11-70s	135-142s	7-57s	93-166s	16-85s	68-90s	20-111s	96-178s	–

Note: DAC, CSIL and P^2IL are not evaluated on certain tasks due to compatibility issues cause by version conflicts. Specifically, the current implementations of DAC and P^2IL are incompatible with the current Gymnasium Robotics suite, while P^2IL and CSIL are incompatible with the Humanoid version used in testing other algorithms.

Learning Stability

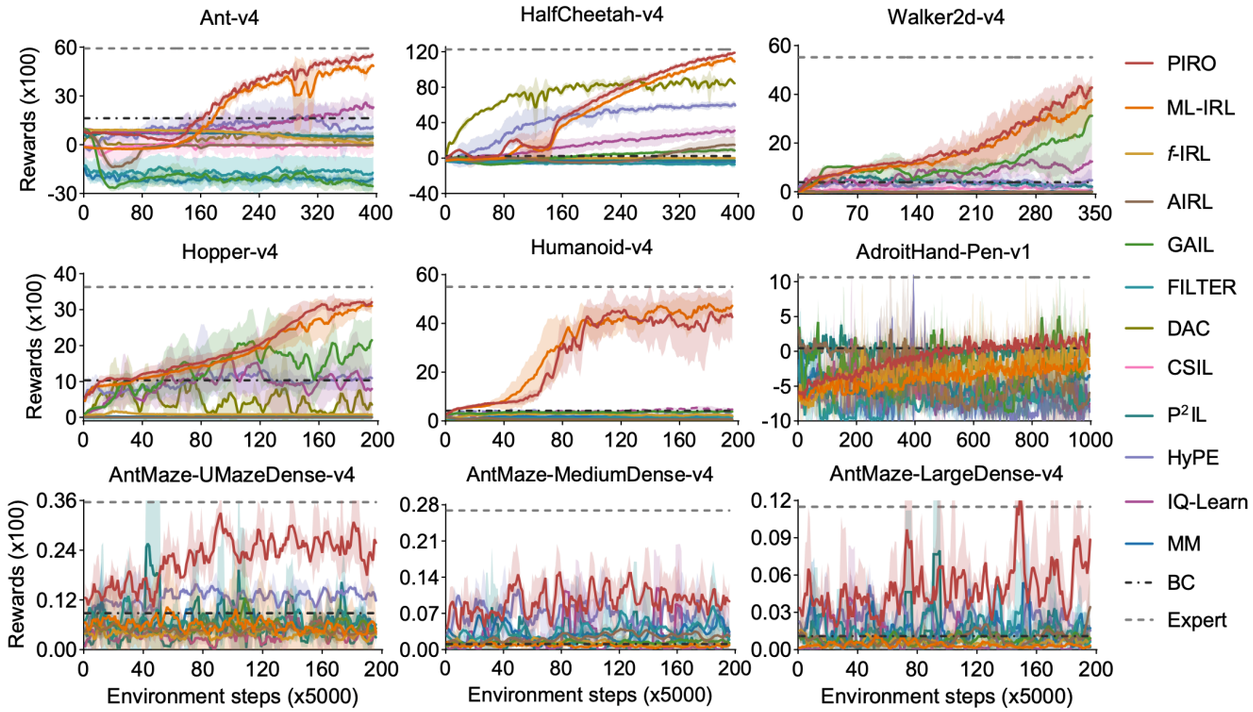


Figure 4: **Reward curves of algorithms** on MuJoCo locomotion tasks and Gym Robotics tasks.

Experimental Evaluation

Reward Recovery and Policy Imitation

Table 1: **Averaged Rewards** (five independent runs) on five MuJoCo and four Gym Robotics tasks.

	Task	Expert	IL		Adv. IRL (Online)				Adv. (Offline)	Non-Adv. Online			Non-Adv. Offline		PIRO	*Gain*
			BC	GAIL	MM	AIRL	FILTER	HyPE	DAC	IQ	ML-IRL	f-IRL	CSIL	P^2IL		
MuJoCo	Ant-v4	5926.2	1631.5	996.9	-304.0	991.4	-376.3	2800.5	923.8	3589.8	5382.5	980.4	420.7	976.6	**5967.2**	+584.7
	Humanoid-v4	5501.0	418.1	508.4	367.2	281.4	291.7	717.5	76.3	1847.5	5573.4	470.4	–	–	**5954.9**	+381.5
	Walker2d-v4	5524.5	384.4	4158.1	70.4	72.8	77.7	1478.7	-3.0	3023.0	4794.7	243.8	686.1	1054.0	**5643.7**	+849.0
	Hopper-v4	3632.8	1034.4	**3535.7**	57.8	13.5	37.3	2593.7	3321.6	3424.5	3316.4	361.7	6.7	25.8	3362.0	-173.7
	Halfcheetah-v4	12266.1	221.2	1298.8	20.3	2251.4	0.3	6473.4	9645.0	3825.5	11873.2	-0.7	-107.2	-0.1	**12587.4**	+714.2
Robotics	AntMaze-UMazeDense-v4	35.6	8.8	5.2	5.1	4.5	6.1	11.9	–	3.9	4.2	3.6	–	3.4	**25.7**	+13.8
	AntMaze-MediumDense-v4	26.9	1.1	1.3	3.4	2.6	1.9	3.0	–	3.4	0.9	1.1	–	2.9	**9.4**	+6.0
	AntMaze-LargeDense-v4	11.5	1.1	0.9	1.7	3.4	0.6	1.5	–	0.8	0.3	0.9	–	0.2	**8.8**	+5.4
	AdroitHandePen-Human-v1	1062.5	44.1	-8.7	-344.3	-593.9	-685.4	-866.7	–	-751.9	-251.2	-65.3	–	-61.2	**254.0**	+209.9
	runtime per iteration	-	-	3-14s	8-79s	5-8s	9-41s	11-70s	135-142s	7-57s	93-166s	16-85s	68-90s	20-111s	96-178s	–

Note: DAC, CSIL and P^2IL are not evaluated on certain tasks due to compatibility issues cause by version conflicts. Specifically, the current implementations of DAC and P^2IL are incompatible with the current Gymnasium Robotics suite, while P^2IL and CSIL are incompatible with the Humanoid version used in testing other algorithms.

Learning Stability

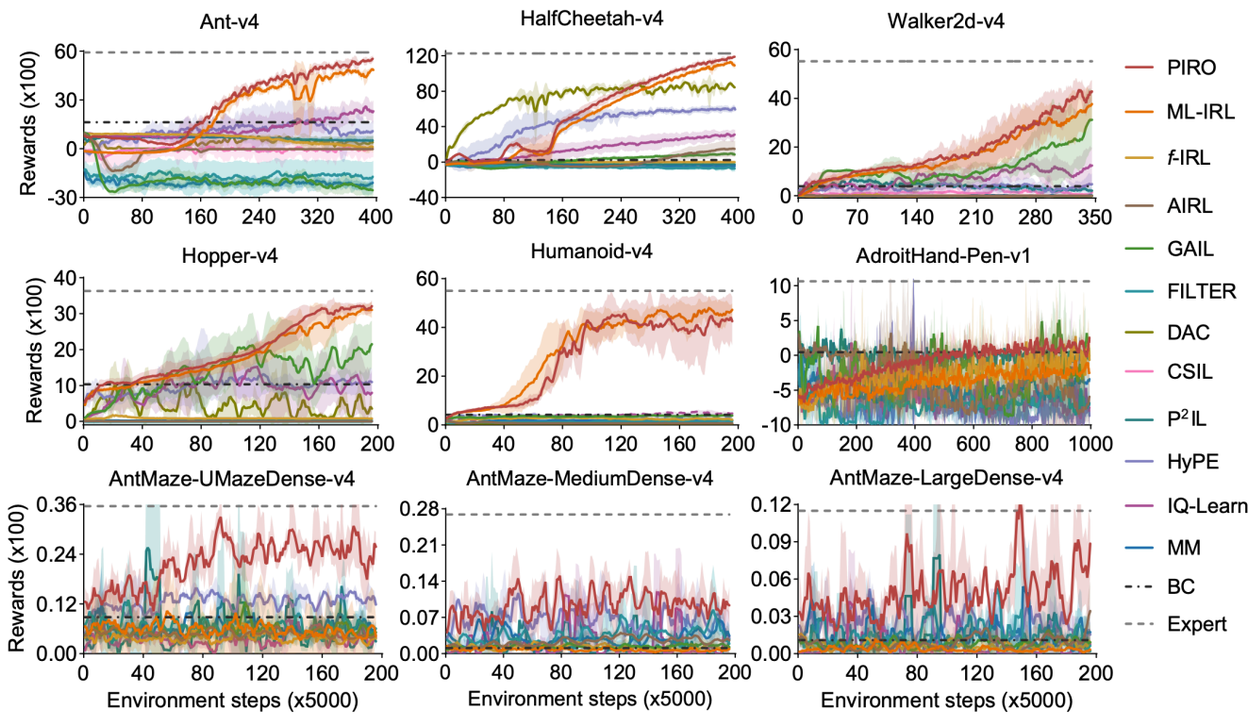


Figure 4: **Reward curves of algorithms** on MuJoCo locomotion tasks and Gym Robotics tasks.

State-Only Rewards

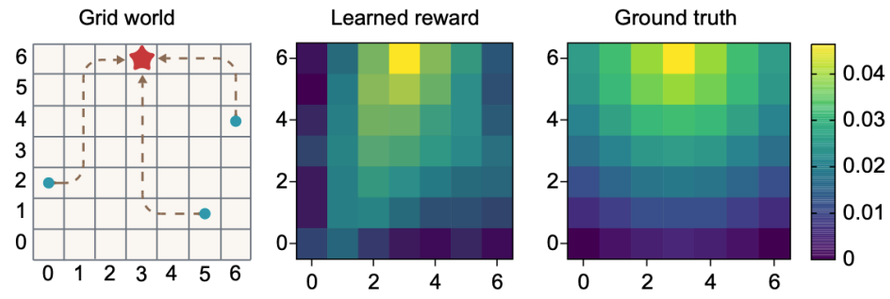


Figure 5: **Experiments on reward recovery in tasks with state-only rewards. Left:** The task is a 7×7 grid world, where the agent starts from a random initial position (blue circles) with the objective of reaching the target position (red star) via the shortest possible path. **Right:** The ground truth reward at each position is defined as the negative Euclidean distance to the terminal state. **Middle:** The reward recovered by PIRO and the ground-truth reward function is highly consistent with the ground truth reward. Cumulative rewards: -9.24 (expert) vs. -8.48 (PIRO).

Experimental Evaluation (con'd)

Reward Transfer

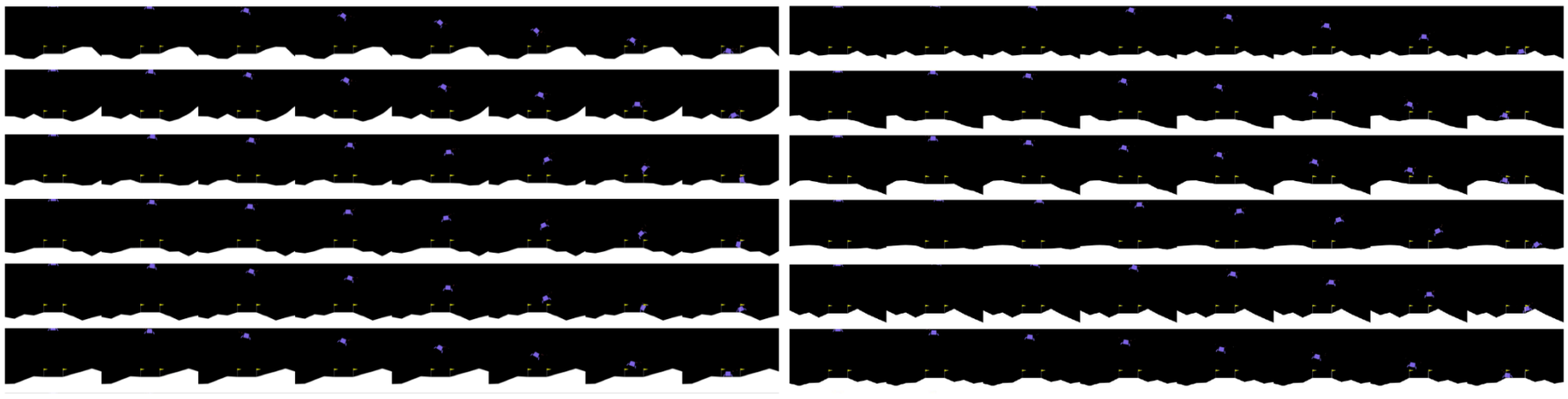


Figure 6: **Results for reward transfer to new environments with altered dynamics. Left panels:** Policy behavior learned by PIRO in the original LunarLander environment. PIRO succeeds in most cases. **Right panels:** Policy behavior under PIRO's learned reward function in LunarLander with altered dynamics (stochastic wind added). The policy is robust in general, despite some failure cases, e.g., row 3.

Experimental Evaluation (con'd)

Reward Transfer

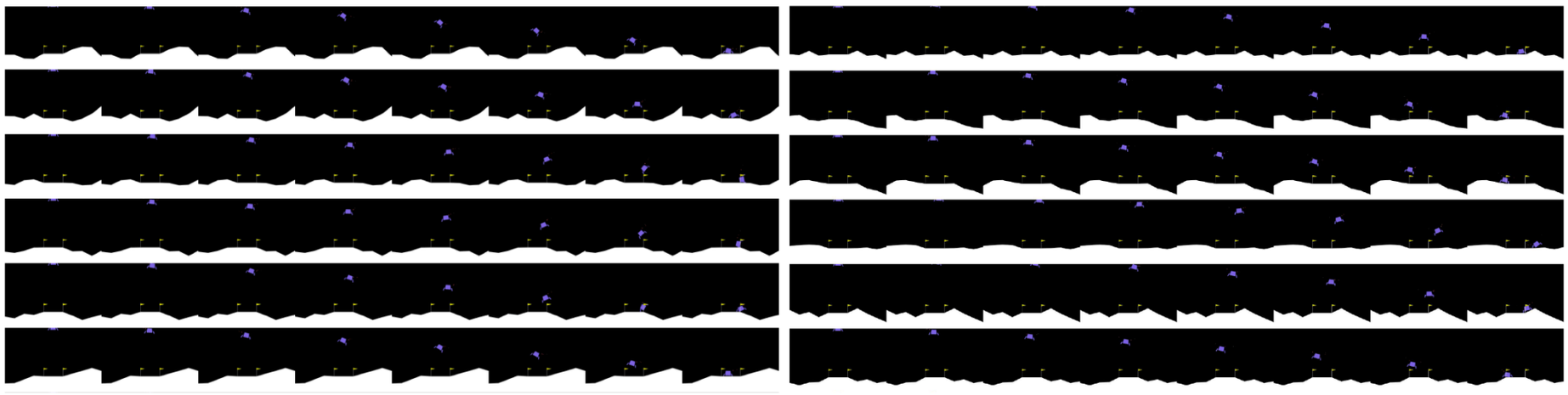


Figure 6: **Results for reward transfer to new environments with altered dynamics. Left panels:** Policy behavior learned by PIRO in the original LunarLander environment. PIRO succeeds in most cases. **Right panels:** Policy behavior under PIRO's learned reward function in LunarLander with altered dynamics (stochastic wind added). The policy is robust in general, despite some failure cases, e.g., row 3.

Real-World Case Study— Animnal Behavior Modeling

Figure 10: Fifteen types of the meerkat behaviors.

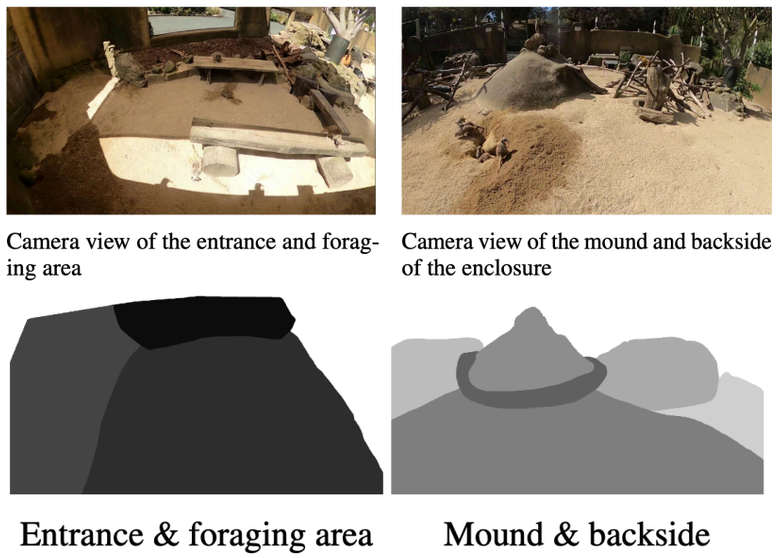


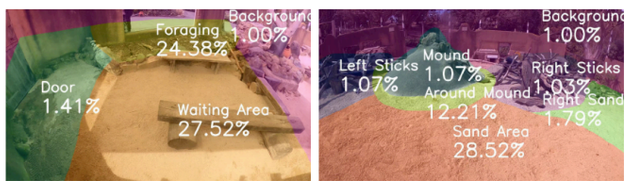
Figure 12: Different colors are labelled for each area to visually illustrate the division of meerkat activity zones.

Experimental Evaluation (con'd)

Real-World Case Study— Animnal Behavior Modeling

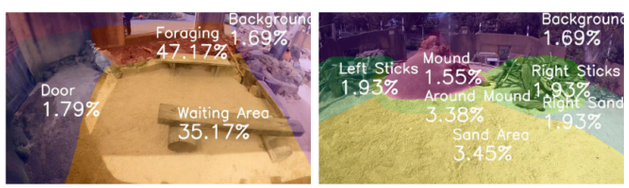
(a) Expert

(b) PIRO (weighted mean error: **5.5%**)

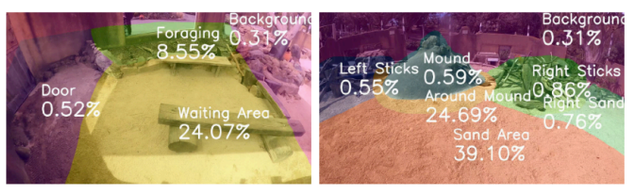


(c) AIRL (weighted mean error: 8.7%)

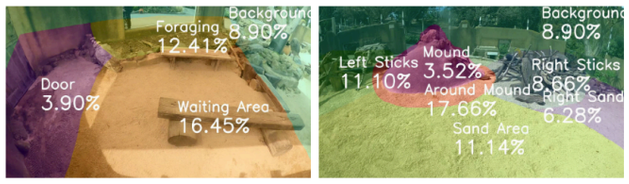
(d) BC (weighted mean error: 9.3%)



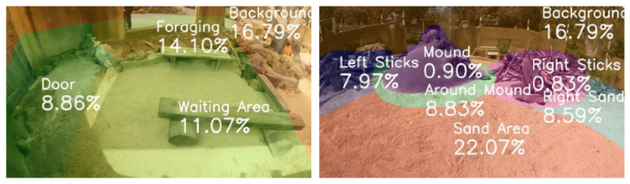
(e) GAIL (weighted mean error: 11.8%)



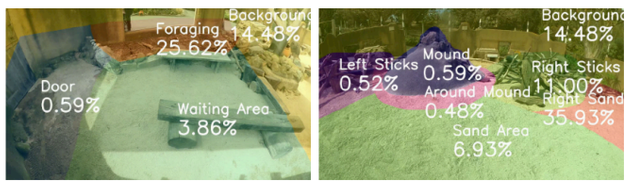
(f) f-IRL (weighted mean error: 16.1%)



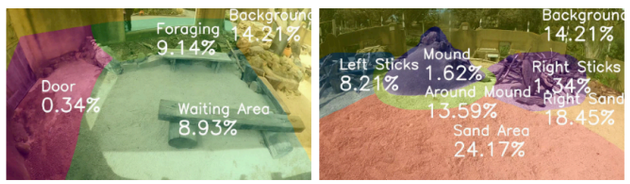
(g) FILTER (weighted mean error: 10.0%)



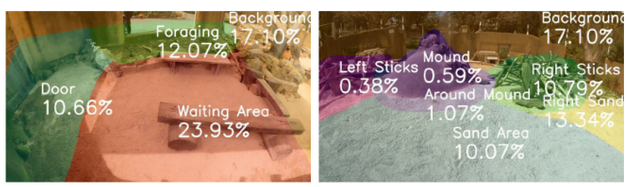
(h) MM (weighted mean error: 10.5%)



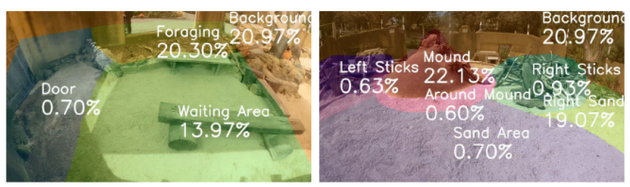
(i) IQ-Learn (weighted mean error: 9.8%)



(j) ML-IRL (weighted mean error: 11.3%)



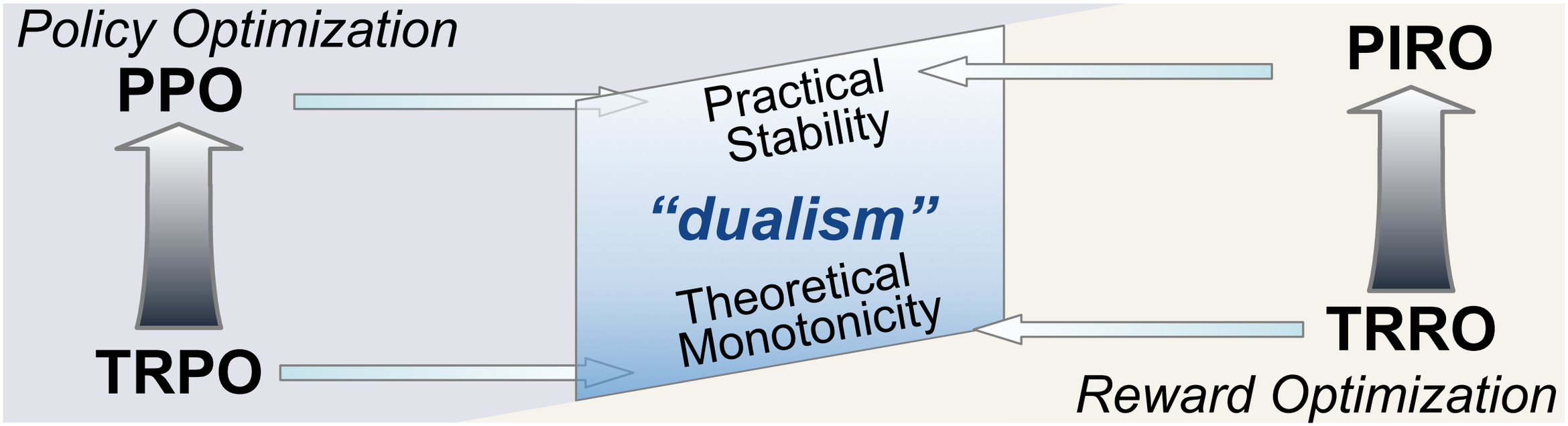
(k) HyPE (weighted mean error: 14.6%)



(l) P^2IL (weighted mean error: 13.0%)

Figure 13: **Regional visitation frequency map** generated by analyzing real meerkat trajectories alongside those produced by algorithms. PIRO achieves the lowest weighted mean error.

Summary



Thanks for your watching !

Scan to View the Project