Reward and Constraint Learning: Foundations for Human-AI Alignment

Human-AI Alignment and Advanced Topics

Sebastian Tschiatschek

Session 4, July 2026

European Summer School on AI (ESSAI) 2026

Course overview

Session 1 (Mon)

- RL Basics
- IRL Basics
- Maximum Entropy IRL

Session 2 (Tue)

- Deep Reward Modeling
- Learning from Human Feedback

Session 3 (Wed)

- Constraints
- Importance and Learning

Session 4 (Fri)

- Human-AI Alignment
- Advanced Topics

Lecture overview

Today's session

- Quick recap
- Constrained Markov Decision Processes (CMDPs)
- Learning with and of constraints
- Cooperative IRL
- IRL with constraints
- Scalable oversight and alignment
- Evaluation and practical pipelines
- Wrap-up, discussion and Q&A

Recap

Session 3 Recap

- Active reward learning
- LLM Alignment via RLHF
- Types of contraints

Constraints for Alignment

What do constraints buy us?

(beyond negative rewards)

Constraints for Alignment

What do constraints buy us?
(beyond negative rewards)

- Guarantees and certifiability
- Separation of concerns (safety + performance)
- Controllability and transparency
- Composability and scalability
- Sample efficiency for known requirements

Formulating Constrained Problems

Key design axes when learning with constraints

Design axis	Common choices / implications
Model knowledge	Known model (MPC/LP/CBF possible); unknown model (model-free RL, model learning + MPC)
Learning-time safety	Violations allowed during learning (optimize in expectation); no violations allowed (safety filters/shields, CBF/MPC, conservative exploration)
Constraint semantics	Hard/pathwise ("always avoid"), cumulative/budget (CMDP), probabilistic/chance, risk (CVaR), robust/worst-case
Enforcement style	Explicit constraint handling (filters, MPC, LP) vs penalty/Lagrangian (learn multipliers λ); both can be combined
Data regime	Online RL (on-policy/off-policy), offline/batch RL (uncertainty-aware, HCOPE), simulators vs. real-world
Objective horizon	Discounted, finite episodic, average-cost (affects LP/occupancy and algorithms)
Specification form	Additive costs vs temporal logic/specs (LTL/STL), equality. vs inequality, time-varying constraints
Observability	Fully observed vs partial (belief-space constraints; chance constraints on belief)
Risk/uncertainty model	Expectation, high-probability, CVaR, distributionally robust (ambiguity sets)
Feasibility	Known feasible baseline vs. unknown; need feasibility checks/slack, fallback policies

Formulating Constrained Problems

Method map: knowledge about the environment $\times$ constraint violations allowed?

	Environment known	Environment unknown
Violations allowed	• Constrained planning/MPC • LP over occupancy (tabular)	• Lagrangian actor-critic • Off-policy constrained RL • Model-based RL
No violations	• Safety filters (CBF/CLF) • Robust/chance-constrained MPC • Reachability/viability	• Shielded RL • Safe exploration • Offline constrained RL

Formulating Constrained Problems

Method map: knowledge about the environment $\times$ constraint violations allowed?

	Environment known	Environment unknown
Violations allowed	• Constrained planning/MPC • LP over occupancy (tabular)	• Lagrangian actor-critic • Off-policy constrained RL • Model-based RL
No violations	• Safety filters (CBF/CLF) • Robust/chance-constrained MPC • Reachability/viability	• Shielded RL • Safe exploration • Offline constrained RL

Too many variants to cover, just picked a few according to my preferences

Constrained Markov Decision Processes

Constrained Markov Decision Processes (CMDPs)

Constrained Markov Decision Process (CMDP)

A Constrained Markov Decision Process is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \{\mathcal{C}\}_{i\in\mathcal{I}}, \{\tau_i\}_{i\in\mathcal{I}}, \rho_0, \gamma)$ with:

- State space $\mathcal{S}$, action space $\mathcal{A}$
- Transition probabilities $\mathcal{P}\colon \mathcal{S}\times\mathcal{A}\to\Delta(\mathcal{S})$
- Reward function $\mathcal{R}\colon \mathcal{S}\times\mathcal{A}\to\mathbb{R}$
- Constraint functions $\mathcal{C}_i\colon \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ and constraint thresholds $\tau_i\in\mathbb{R}_{\geq 0}$ for $i\in\mathcal{I}$
- Initial state distribution ρ_0, discount factor γ

Variants: Episodic/average-cost; transition & costs $\mathcal{R}\colon \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}\to R, \mathcal{C}_i\colon \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}$

Altman, E. (1995). *Constrained Markov decision processes*.

Constrained Markov Decision Processes (CMDPs)

Planning in CMDPs

Want to find optimal policy satisfying the constraints:

$$\pi^* = \arg\max_{\pi} \ \mathbb{E}_{\pi,\mathcal{P}}\Big[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)\Big]$$

$$\text{s.t.} \ \ \mathbb{E}_{\pi,\mathcal{P}}\Big[\sum_{t=0}^{\infty} \gamma^t \mathcal{C}_i(s_t, a_t)\Big] \leq \tau_i \quad \forall i \in [\mathcal{I}]$$

Constrained Markov Decision Processes (CMDPs)

Planning in CMDPs

Want to find optimal policy satisfying the constraints:

$$\pi^* = \arg\max_{\pi} \ \mathbb{E}_{\pi,\mathcal{P}}\Big[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)\Big]$$

$$\text{s.t.} \ \ \mathbb{E}_{\pi,\mathcal{P}}\Big[\sum_{t=0}^{\infty} \gamma^t \mathcal{C}_i(s_t, a_t)\Big] \leq \tau_i \quad \forall i \in [\mathcal{I}]$$

Are CMDPs as easy as MDPs?

What challenges could arise?

Noteworthy aspects of CMDPs

- In general there is no deterministic optimal policy but stationary randomized ones suffice
- There is not a single Bellman optimality operator unless you fix Lagrange multipliers λ resulting in a standard MDP with augmented reward:
  - Optimize via Lagrangian or LP over occupancy measures; strong duality + KKT
  - Variants of policy gradient (constrained policy gradient) are available

Planning in CMDPs via Occupancy Measures

Develop a linear program over discounted state-action visitation frequencies for solving CMDPs

Discounted occupancy measure for a finite MDP, discounted, with initial state distribution ρ_0:

$$x^{\pi}(s,a) = \sum_{t=0}^{\infty} \gamma^t P_{\pi,\mathcal{P}}(s_t = s, a_t = a \mid s_0 \sim \rho_0)$$

- Flow conservation (for all $s \in \mathcal{S}$):

$$\sum_a x(s,a) = \rho_0(s) + \gamma \sum_{s',a'} P(s|s',a')x(s',a')$$

- Nonnegativity: $x(s,a) \geq 0$

Planning in CMDPs via Occupancy Measures

Expected discounted return/costs become linear in x:

$$\mathbb{E}_{\pi,\mathcal{P}}\left[\sum_t \gamma^t \mathcal{R}(s_t, a_t)\right] = \sum_{s,a} x(s,a) r(s,a)$$

$$\mathbb{E}_{\pi,\mathcal{P}}\left[\sum_t \gamma^t \mathcal{C}_i(s_t, a_t)\right] = \sum_{s,a} x(s,a) c_i(s,a)$$

LP formulation for CMDPs and Policy Recovery

Primal LP

$$\max_{x \geq 0} \quad \sum_{s,a} x(s,a)\mathcal{R}(s,a)$$

$$\text{s.t.} \quad \sum_{a} x(s,a) = \rho_0(s) + \gamma \sum_{s',a'} P(s|s',a')x(s',a') \quad \forall s$$

$$\sum_{s,a} x(s,a)\mathcal{C}_i(s,a) \leq \tau_i \quad \forall i \in \mathcal{I}$$

LP formulation for CMDPs and Policy Recovery

Primal LP

$$\begin{aligned} \max_{x \geq 0} \quad & \sum_{s,a} x(s,a)\mathcal{R}(s,a) \\ \text{s.t.} \quad & \sum_{a} x(s,a) = \rho_0(s) + \gamma \sum_{s',a'} P(s|s',a')x(s',a') \quad \forall s \\ & \sum_{s,a} x(s,a)\mathcal{C}_i(s,a) \leq \tau_i \quad \forall i \in \mathcal{I} \end{aligned}$$

Extracting a stationary randomized policy:

$$\pi(a|s) = \begin{cases} \dfrac{x(s,a)}{\sum_{a'} x(s,a')} & \text{if } \sum_{a'} x(s,a') > 0 \\ \text{arbitrary distribution over } \mathcal{A} & \text{otherwise} \end{cases}$$

Exact in the tabular discounted setting: returns an optimal occupancy $x^\star$ and a stationary policy $\pi^\star$ satisfying the constraints

Dual, Interpretation, and Practical Notes

Dual (sketch; variables $V(s)$ for flow constraints, $\lambda_i \geq 0$ for costs):

$$\min_{V,\lambda \geq 0} \quad \sum_s \rho_0(s) V(s) + \sum_i \lambda_i \tau_i$$

$$\text{s.t.} \quad V(s) \geq \mathcal{R}(s,a) - \sum_i \lambda_i \mathcal{C}_i(s,a) + \gamma \sum_{s'} P(s'|s,a) V(s') \quad \forall (s,a)$$

Interpretation:

- λ are Lagrange multipliers (penalties) for constraints
- the constraint becomes an augmented reward $\mathcal{R}_\lambda = \mathcal{R} - \sum_i \lambda_i \mathcal{C}_i$
- V is a value function for the augmented MDP

Solvable with standard LP solvers for moderate-size tabular problems

Violations Allowed But Environment is Unknown

Lagrangian RL: Problem Formulation

Objective (CMDP)

$$\max_{\pi} J_r(\pi) \quad \text{s.t.} \quad J_c(\pi) \leq \tau$$

where J_r is the expected return, J_c is the expected cost, and τ is the safety budget

Lagrangian function. Convert the constrained problem into an unconstrained **min-max** problem:

$$\min_{\lambda \geq 0} \max_{\pi} L(\pi, \lambda) := J_r(\pi) - \lambda \left(J_c(\pi) - \tau \right)$$

Dual variable (λ)

- $\lambda \geq 0$ acts as a "penalty weight" for safety violations
- High $\lambda \rightarrow$ High priority on safety (Cost reduction)
- Low $\lambda \rightarrow$ High priority on performance (Reward maximization)

Lagrangian Actor-Critic (LAC): Update Rules

LAC uses a primal-dual approach, where the Actor (Primal) and the Multiplier (Dual) are updated on different time-scale

1. Actor update (primal).

$$\theta \leftarrow \theta + \alpha_\theta \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a|s)(Q_r(s,a) - \lambda Q_c(s,a))\right]$$

2. Multiplier update (dual).

$$\lambda \leftarrow \max\left(0,\ \lambda + \alpha_\lambda \left(J_c(\pi) - \tau\right)\right)$$

Two-time-scale dynamics

- **Critics:** Estimate Q_r and Q_c (fastest scale)
- **Actor:** Updates policy to satisfy current λ (medium scale)
- **Lagrangian (λ):** Adjusts slowly to satisfy the long-term constraint (slowest scale)

Limitations of Lagrangian RL

- Overshooting
  - The agent must explore and violate constraints to learn the penalty
  - Violation is inevitable during initial training phases
- Numerical stability & hyperparameters
  - Typically, high sensitivity to the dual learning rate (α_λ)
  - If τ (budget) is unachievable, λ can explode, leading to training divergence
- Non-stationarity
  - Objective for the Actor changes whenever λ is updated, making the learning environment non-stationary and harder to converge

Learning About Constraints

Learning Constraints Offline: CoCoRL

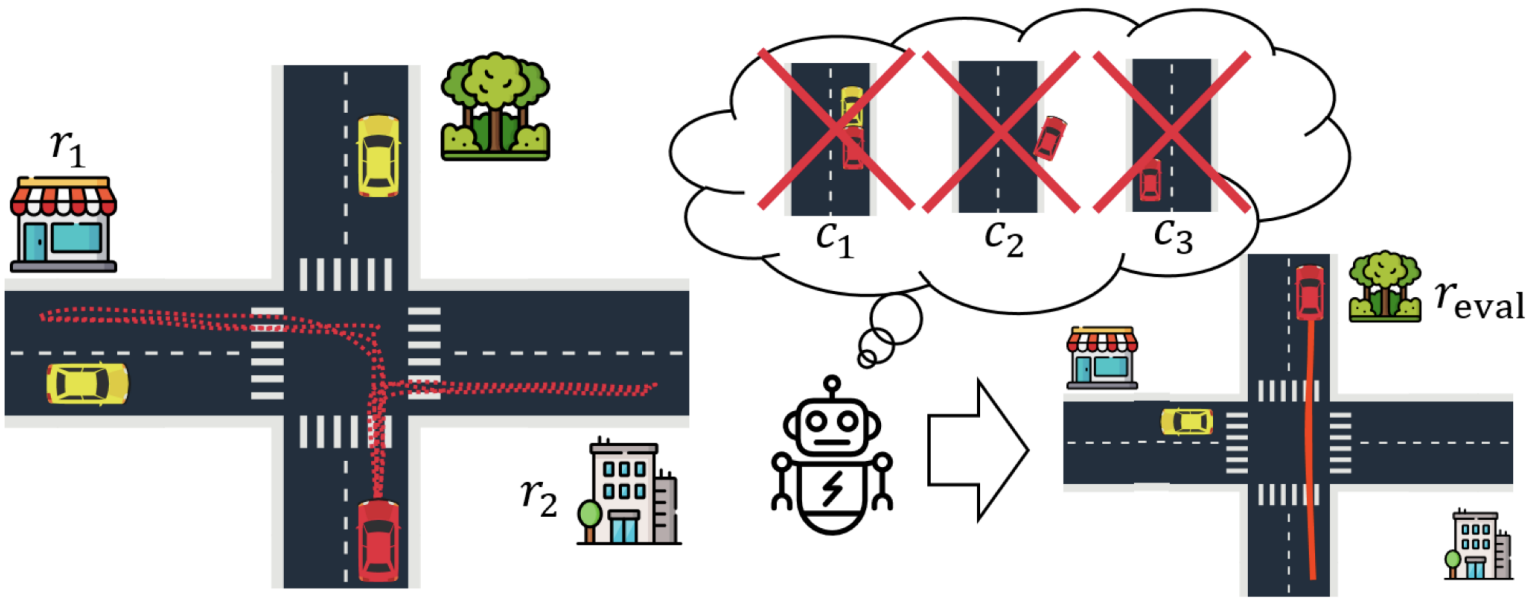
Learning safety constraints from demonstrations with unknown (individual) rewards

Our CoCoRL framework introduces:

- Learn safety constraints from heterogeneous human demonstrations with differing reward functions
- Challenge: Demonstrators are optimizing different, unknown task rewards, but they all respect the same underlying safety constraints
- Solution: Convex Constraint Representation Learning (CoCoRL) separates individual task preferences to identify shared safety constraints.

Lindner, D., Chen, X., et al. (2024). *Learning safety constraints from demonstrations with unknown rewards*, *International Conference on Artificial Intelligence and Statistics*.

Learning Constraints Offline: CoCoRL



Learning Constraints Offline: CoCoRL

Setting

- Initial CMDP without reward and constraints functions is given
- Constraints are linear in some features (resp. their discounted expectation), i.e., $\mathcal{C}_j(s,a) = \langle \mathbf{w}, \mathbf{f}(s,a) \rangle$ and we require $\mathcal{C}_j(s,a) \leq \xi_j$
- We have demonstrations for different rewards satisfying the constraints
- For testing, we receive a reward function and must identify a safe policy

CoCoRL: Algorithm

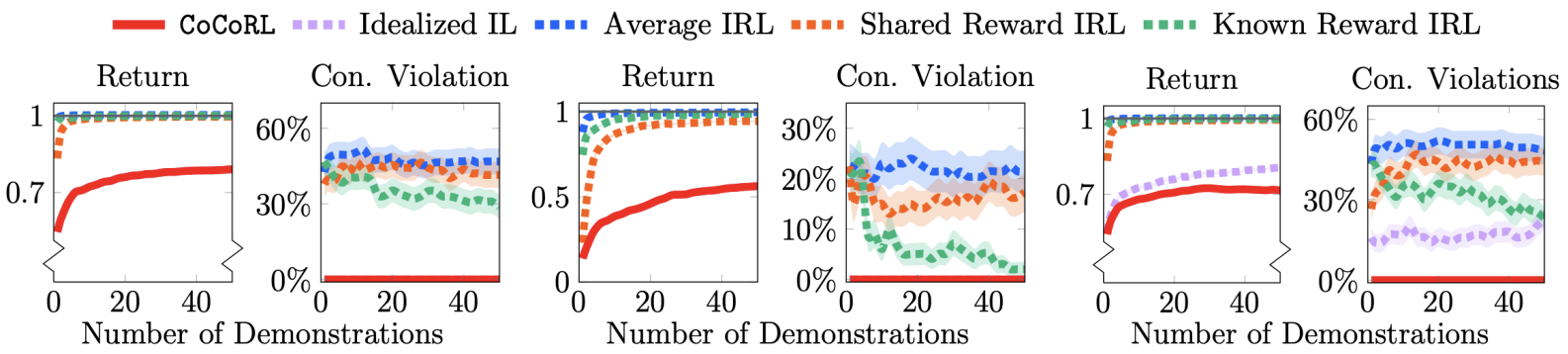
If a state-action pair (s, a) is never visited by any expert (despite different task goals), it is likely because it is violates a constraint

Algorithm

1. Demonstration processing: Map all expert demonstrations $\{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$ into a shared feature space
2. Constraint inference: Seperate "visited" (safe) regions from the "avoided" (unsafe) regions
3. Safe set construction ($\mathcal{S}$): Define $\mathcal{K} = \{(s, a) \mid \langle \mathbf{w}_{inferred}, \mathbf{f}(s, a) \rangle \leq \xi\}$
4. Policy synthesis: Optimize π in the environment while restricted to the subspace defined by $\mathcal{K}$

CoCoRL: Performance

Theoretical guarantees + strong empirical performance



(a) Single environment. (b) Transfer to new task. (c) Transfer to new environment.

Safe Exploration

Safe Exploration: No-Violation Setting

Learn an optimal policy π in an unknown environment while ensuring $s_t \in \mathcal{S}_{safe}, \forall t$

Feasibility requirements:

- To guarantee safety, we cannot explore blindly
- Fundamental assumption: Need *Safety Prior* (e.g., known safe controller, coarse model, or pre-defined safe set $\mathcal{S}_{safe}$)

Mechanisms for feasibility:

1. Shielding: A symbolic/logic monitor that overrides unsafe actions
2. CBF-Gating: Using *control barrier functions* to project unsafe actions into the safe action set $\mathcal{A}_{safe}(s)$
3. Uncertainty awareness: Only explore states where the model's confidence is high enough to certify safety

Foundations for Safe Exploration

The core challenge: In the no-violation setting, we cannot explore blindly and must rely on a "safety contract" between the agent and the environment:

- Safe initialization: Access to a known safe state s_0 and a base controller π_{safe} that keeps the agent within a verifiable safe set $\mathcal{S}_{safe}$
- Dynamics & Lipschitz continuity: We model the environment dynamics $f(s, a)$ and assume Lipschitz continuity ($|f(s_1) - f(s_2)| \leq L|s_1 - s_2|$)
  - L allows us to calculate a **safety buffer** (margin) to handle model drift
- Uncertainty awareness: The agent must quantify model error $e(s, a)$ (e.g., via GPs)

What is $h(s)$? The Geometry of Safety

To keep the system safe, we define a barrier function $h(s)$ that acts as a scalar proxy for safety

- Safe set ($\mathcal{S}$): Defined by the set of states where $h(s) \geq 0$.
- Boundary: Region where $h(s) = 0$
- Goal: Need to ensure that if $h(s_0) \geq 0$, then $h(s_t) \geq 0$ for all $t > 0$

Example: If safety means staying within a distance d of a wall, we can define $h(s) = d - \text{dist}(s, \text{wall})$.

- When dist $< d$, $h(s) > 0$ (safe)
- When dist $= d$, $h(s) = 0$ (on the boundary)

Algorithm: Safe CBF-RL

Wrap the RL policy π_θ in a QP-based safety filter

CBF Projection Filter (assuming deterministic transitions)

$$a^* = \arg\min_a \|a - a_{raw}\|^2$$

$$\text{s.t. } \underbrace{\mathbb{E}_{s' \sim \mathcal{P}(s'|s,a)}[h(s') - h(s)]}_{\text{often must be linearized}} \geq -\gamma h(s) + L \cdot \underbrace{\sigma(s,a)}_{\text{model uncertainty}}$$

Iterate

1. **Action:** Actor proposes $a_{raw} = \pi_\theta(s)$
2. **Filter:** Solve the QP to find the safe action a^* nearest to a_{raw}
3. **Execute:** Deploy a^* and observe (s', r, c)
4. **Update:** Update π_θ via RL and refine dynamics model and uncertainty $\sigma(s, a)$

Cooperative Inverse Reinforcement Learning

Alignment Challenge

How do we ensure AI systems act in accordance with human values when the objective function is not fully known?

So far we had roughly...

- Standard RL: Assumes the reward function $\mathcal{R}$ is fixed and known
- IRL: AI learns $\mathcal{R}$ by observing expert human behavior

Alignment Challenge

How do we ensure AI systems act in accordance with human values when the objective function is not fully known?

So far we had roughly...

- Standard RL: Assumes the reward function $\mathcal{R}$ is fixed and known
- IRL: AI learns $\mathcal{R}$ by observing expert human behavior

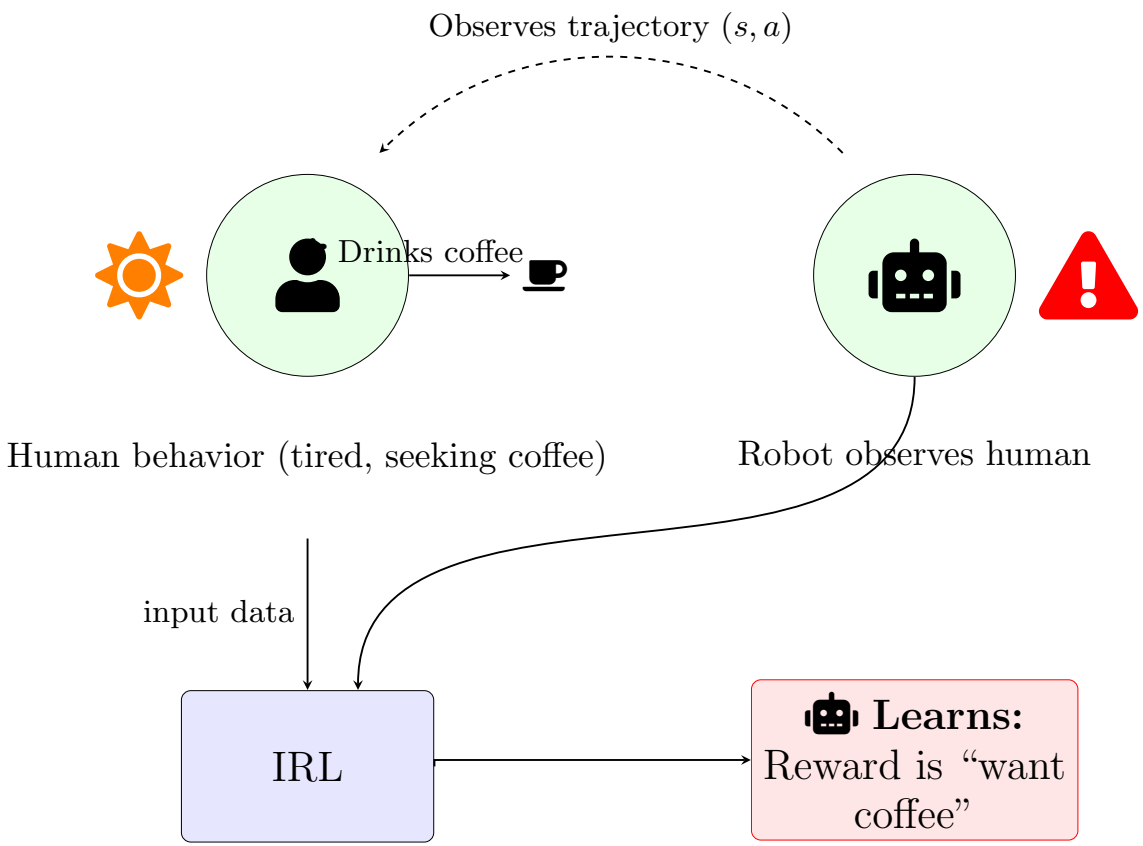
Limitation of standard IRL

Assumes the human is an expert and the AI is a passive observer

Alignment Challenge

What should an agent do?

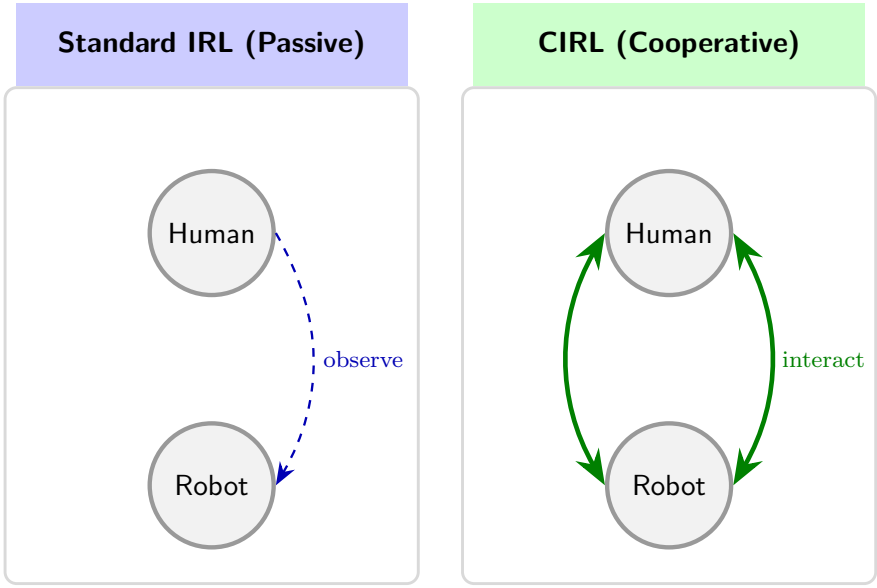
We do not necessarily want an AI agent to directly adopt a human's reward function



Cooperative IRL (CIRL)

Hadfield-Menell, D. et al. (2016). *Cooperative inverse reinforcement learning*, *NeurIPS*.

- Shifts from *passive observation* to *active cooperation*
- Two-agent game (Human H, Robot R)
- Both agents want to maximize the same reward $\mathcal{R}$ (the human's reward), but only H knows $\mathcal{R}$



Robot as Observer
Human is passive

Robot as apprentice
Active inquiry/influence

Formal Definition of CIRL

Cooperative Inverse Reinforcement Learning (CIRL) game

$$\mathcal{M}_{\text{CIRL}} = \langle \mathcal{S}, \{\mathcal{A}_H, \mathcal{A}_R\}, \mathcal{P}, \Theta, \mathcal{R}, P_0, \gamma \rangle$$

- $\mathcal{S}$ is the set of world states
- $\mathcal{A}_H, \mathcal{A}_R$ are action spaces for the human and robot
- $\mathcal{P}$ is the transition function $\mathcal{P}(s'|s, a_H, a_R)$
- Θ is the set of reward parameters (observed only by the human)
- $\mathcal{R}(s, a_H, a_R; \theta)$ is the shared reward function
- P_0 is the distribution over the initial state $P(s_0, \theta)$
- γ is the discount factor

CIRL Interaction Protocol

Behavior defined by a pair of policies (π_H, π_R):

- **Initialization:** $(s_0, \theta) \sim P_0$ is sampled; H observes θ; robot R does not
- **Each timestep t:**
  1. Agents observe state s_t
  2. H and R select actions a_H^t, a_R^t
  3. Both receive shared reward $r_t = \mathcal{R}(s_t, a_H^t, a_R^t; \theta)$ and observe each others actions
  4. Next state is sampled: $s_{t+1} \sim P_T(s' | s_t, a_H^t, a_R^t)$.

Formalizing CIRL

$$V^*(\theta) = \max_{\pi_R} \mathbb{E}_{\pi_H, \pi_R} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t, a_t^H, \theta) \mid \theta \right]$$

The robot's policy must account for the human's optimal behavior π_H to optimize its own performance

- The robot "realizes" its actions serve two purposes:
  1. Exploitation: Achieving the goal
  2. Exploration/inference: Gathering information to learn the reward parameters θ
- The human "realizes" the robot is learning, so the human should behave pedagogically (teach the robot)

Theoretical Foundation of CIRL

Theorem

Let $\mathcal{M}$ be an arbitrary CIRL game with state space $\mathcal{S}$ and reward space Θ. There exists a (single-actor) POMDP $\mathcal{M}_C$ with (hidden) state space $\mathcal{S}_C$ such that $|\mathcal{S}_C| = |\mathcal{S}| \cdot |\Theta|$ and, for any policy pair in $\mathcal{M}$, there is a policy in $\mathcal{M}_C$ that achieves the same sum of discounted rewards.

Theoretical Foundation of CIRL

Theorem

Let $\mathcal{M}$ be an arbitrary CIRL game with state space $\mathcal{S}$ and reward space Θ. There exists a (single-actor) POMDP $\mathcal{M}_C$ with (hidden) state space $\mathcal{S}_C$ such that $|\mathcal{S}_C| = |\mathcal{S}| \cdot |\Theta|$ and, for any policy pair in $\mathcal{M}$, there is a policy in $\mathcal{M}_C$ that achieves the same sum of discounted rewards.

- R's belief about θ is a sufficient statistic for optimal behavior
- For any CIRL problem, there exists an optimal policy pair (π_H^*, π_R^*) that only depends on the current state and R's belief

Robot as Apprentice: Bayesian Inference

- Formulate apprenticeship learning as turn-based CIRL with a learning phase and a deployment phase
- As the interaction progresses, the robot updates its belief about the goal θ based on human actions a_H

$$P(\theta \mid a_H, s) \propto P(a_H \mid \theta, s) P(\theta)$$

- An "expert" (own) reward maximizing demonstration might not account for that actions impact on R's belief

Theorem: Suboptimality of Isolation

Isolation Assumption. In standard IRL, we assume the human acts as if the robot is not present:

$$\pi_H^{iso} = \arg\max_{\pi_H} V(\pi_H, \emptyset; \theta)$$

Theorem (Hadfield-Menell et al., 2016)

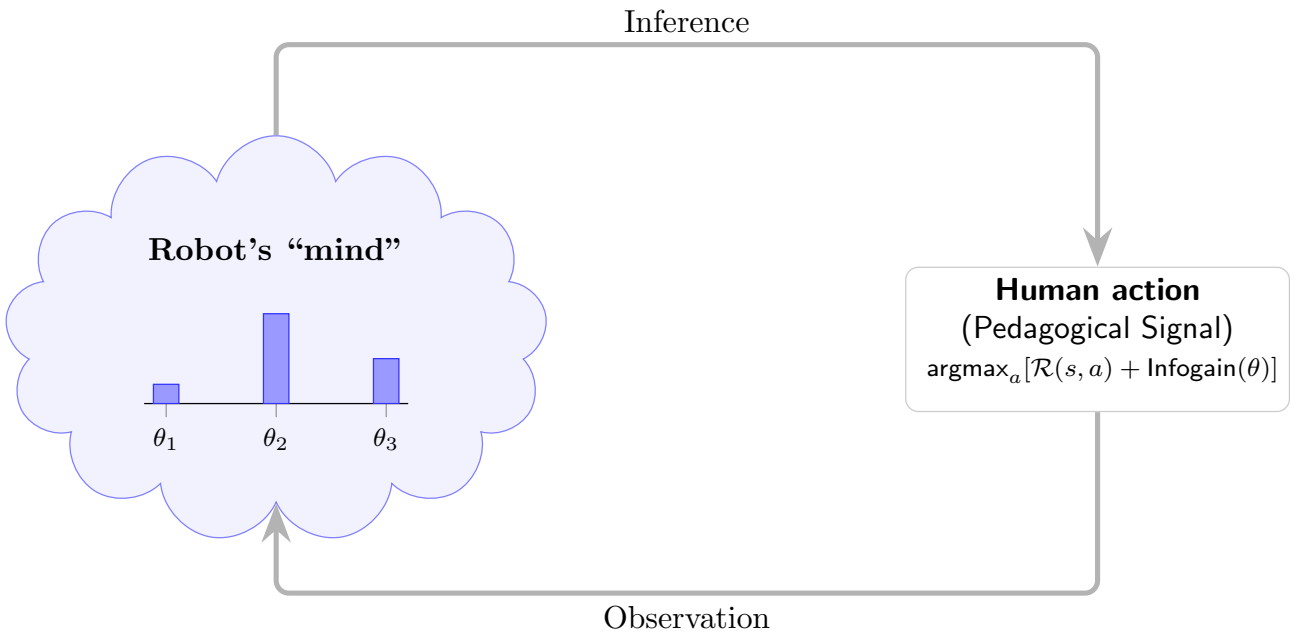
Isolation is strictly suboptimal in CIRL settings where coordination requires information transfer:

$$V(\pi_H^*, \pi_R^*; \theta) \geq V(\pi_H^{iso}, \pi_R^*; \theta)$$

- Coordination requires *pedagogical* action
- The human should trade off immediate reward to reduce the robot's uncertainty about θ
- Turns the alignment problem into a **communication game**

Pedagogical Behavior

- In standard IRL, the human ignores the robot
- In CIRL, the human intentionally acts to help the robot disambiguate the reward function
- This turns the interaction into an active learning problem for the robot and a teaching problem for the human



Implications for Alignment

- **Humility:** Because the robot is uncertain about θ, it naturally exhibits humility; it does not blindly follow a potentially flawed, mis-specified proxy objective
- **Safety:** The robot is less likely to take extreme actions because it wants to avoid high-regret outcomes if it has misunderstood the human's reward

Example Results

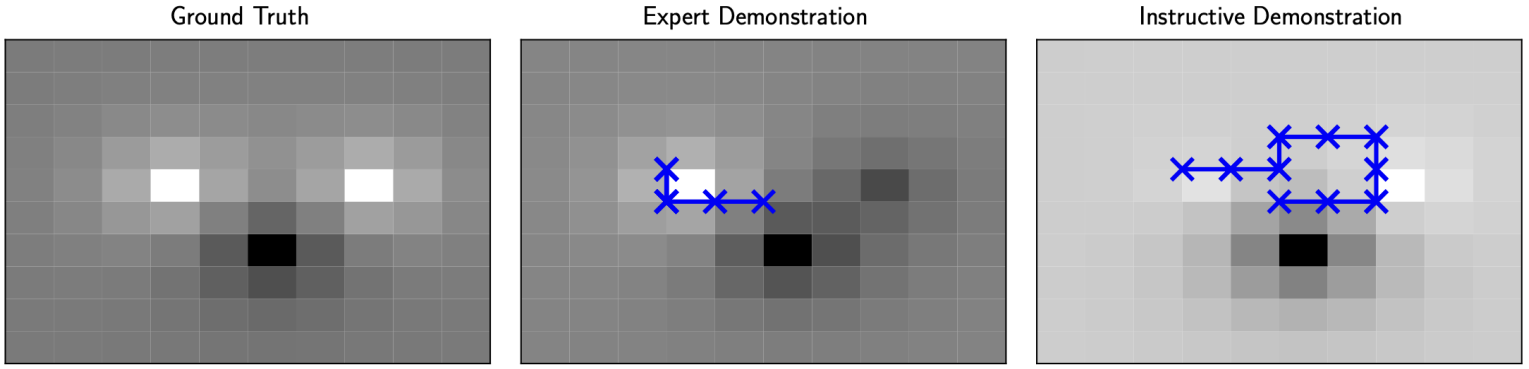


Figure 1: The difference between demonstration-by-expert and instructive demonstration in the mobile robot navigation problem from Section 4. Left: The ground truth reward function. Lighter grid cells indicates areas of higher reward. Middle: The demonstration trajectory generated by the expert policy, superimposed on the maximum a-posteriori reward function the robot infers. The robot successfully learns where the maximum reward is, but little else. Right: An instructive demonstration generated by the algorithm in Section 3.4 superimposed on the maximum a-posteriori reward function that the robot infers. This demonstration highlights both points of high reward and so the robot learns a better estimate of the reward.

Hadfield-Menell, D. et al. (2016). *Cooperative inverse reinforcement learning*, *NeurIPS*.

Key Insights

1. Alignment is a coordination problem, not just an observation problem
2. Uncertainty about objectives is a feature, not a bug — it promotes safety and humility
3. Humans and AI must interact to build shared intent

Learner-aware Teaching: Inverse Reinforcement Learning with Preferences and Constraints

How well can IRL work under mismatch and constraints?

[NeurIPS'19]

Mismatch: Learner does not "blindly" follow demonstrations

How well can IRL work under mismatch and constraints?

[NeurIPS'19]

Mismatch: Learner does not "blindly" follow demonstrations

Learner's model:

How well can IRL work under mismatch and constraints?

[NeurIPS'19]

Mismatch: Learner does not "blindly" follow demonstrations

Learner's model:

- Learner has preferences / constraints ($\phi_c(s_t)$)

How well can IRL work under mismatch and constraints?

[NeurIPS'19]

Mismatch: Learner does not "blindly" follow demonstrations

Learner's model:

- Learner has preferences / constraints ($\phi_c(s_t)$)
- Learner identifies $\hat{\mathbf{w}}$ / π^L via

$$\begin{aligned} \underset{\mathbf{w}}{\text{argmax}} \quad & H(\pi(\mathbf{w})) - C_r \cdot \|\delta_r^{\text{soft}}\|_p - C_c \cdot \|\delta_c^{\text{soft}}\|_p \\ \text{s.t.} \quad & |\mu_r(\pi(\mathbf{w}))[i] - \mu_r(\pi^T)[i]| \leq \delta_r^{\text{hard}}[i] + \delta_r^{\text{soft}}[i] \\ & g_j(\mu_c(\pi(\mathbf{w}))) \leq \delta_c^{\text{hard}}[j] + \delta_c^{\text{soft}}[j] \end{aligned}$$

How well can IRL work under mismatch and constraints?

[NeurIPS'19]

Mismatch: Learner does not "blindly" follow demonstrations

Learner's model:

- Learner has preferences / constraints ($\phi_c(s_t)$)
- Learner identifies $\hat{\mathbf{w}}$ / π^L via

$$\underset{\mathbf{w}}{\text{argmax}} \quad H(\pi(\mathbf{w})) - C_r \cdot \|\delta_r^{\text{soft}}\|_p - C_c \cdot \|\delta_c^{\text{soft}}\|_p$$

$$\text{s.t.} \quad |\mu_r(\pi(\mathbf{w}))[i] - \mu_r(\pi^T)[i]| \leq \delta_r^{\text{hard}}[i] + \delta_r^{\text{soft}}[i]$$

$$g_j(\mu_c(\pi(\mathbf{w}))) \leq \delta_c^{\text{hard}}[j] + \delta_c^{\text{soft}}[j]$$

How well can IRL work under mismatch and constraints?

[NeurIPS'19]

Mismatch: Learner does not "blindly" follow demonstrations

Learner's model:

- Learner has preferences / constraints ($\phi_c(s_t)$)
- Learner identifies $\hat{\mathbf{w}}$ / π^L via

$$\underset{\mathbf{w}}{\text{argmax}} \quad H(\pi(\mathbf{w})) - C_r \cdot \|\delta_r^{\text{soft}}\|_p - C_c \cdot \|\delta_c^{\text{soft}}\|_p$$

$$\text{s.t.} \quad |\mu_r(\pi(\mathbf{w}))[i] - \mu_r(\pi^T)[i]| \leq \delta_r^{\text{hard}}[i] + \delta_r^{\text{soft}}[i]$$

$$g_j(\mu_c(\pi(\mathbf{w}))) \leq \delta_c^{\text{hard}}[j] + \delta_c^{\text{soft}}[j]$$

How well can IRL work under mismatch and constraints?

[NeurIPS'19]

- What does the model do?
- For $C_r, C_c \to \infty$, $\delta_r^{\text{hard}} = \delta_c^{\text{hard}} = 0$, learner's model is

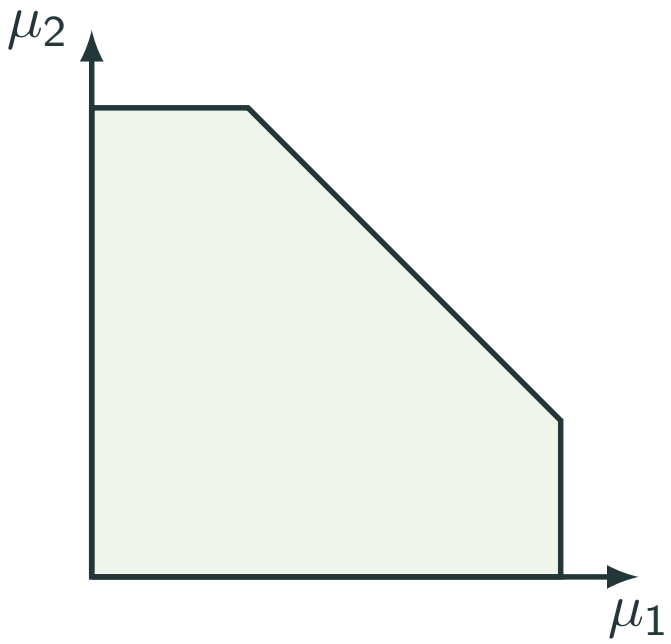
$$\begin{aligned} \underset{\mathbf{w}}{\text{argmin}} \quad & \|\mu_r(\pi(\mathbf{w})) - \mu_r(\pi^T)\|_2 \\ \text{s.t.} \quad & g_j(\mu_c(\pi(\mathbf{w}))) \leq 0 \end{aligned}$$

How well can IRL work under mismatch and constraints?

[NeurIPS'19]

- What does the model do?
- For $C_r, C_c \to \infty$, $\delta_r^{\text{hard}} = \delta_c^{\text{hard}} = 0$, learner's model is

$$\begin{aligned} \underset{\mathbf{w}}{\text{argmin}} \quad & \|\mu_r(\pi(\mathbf{w})) - \mu_r(\pi^T)\|_2 \\ \text{s.t.} \quad & g_j(\mu_c(\pi(\mathbf{w}))) \leq 0 \end{aligned}$$

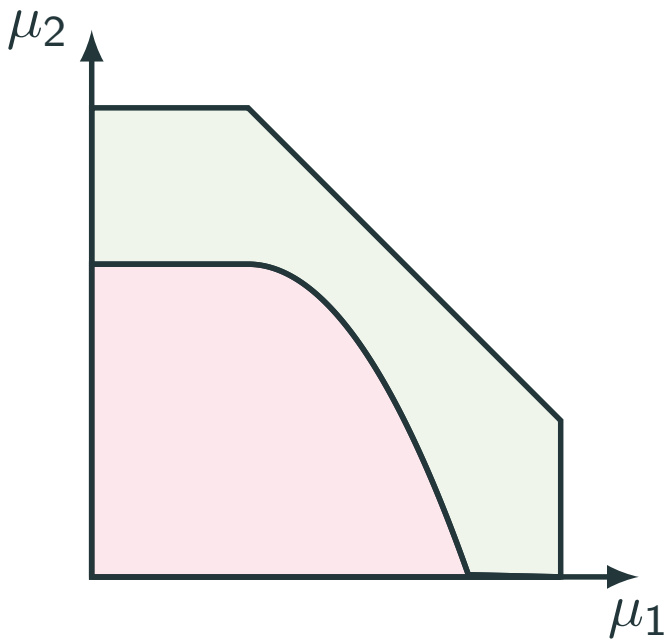


How well can IRL work under mismatch and constraints?

[NeurIPS'19]

- What does the model do?
- For $C_r, C_c \to \infty$, $\delta_r^{\text{hard}} = \delta_c^{\text{hard}} = 0$, learner's model is

$$\begin{aligned} \underset{\mathbf{w}}{\text{argmin}} \quad & \|\mu_r(\pi(\mathbf{w})) - \mu_r(\pi^T)\|_2 \\ \text{s.t.} \quad & g_j(\mu_c(\pi(\mathbf{w}))) \leq 0 \end{aligned}$$

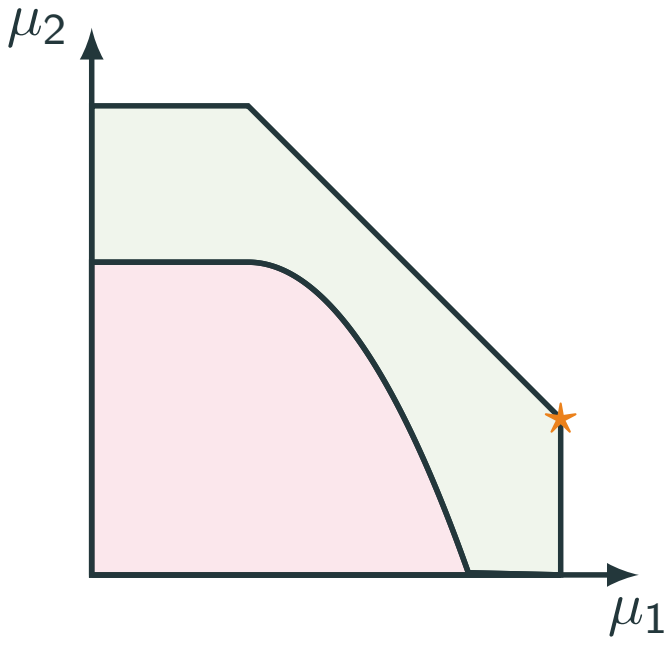


How well can IRL work under mismatch and constraints?

[NeurIPS'19]

- What does the model do?
- For $C_r, C_c \to \infty$, $\delta_r^{\text{hard}} = \delta_c^{\text{hard}} = 0$, learner's model is

$$\begin{aligned} \underset{\mathbf{w}}{\text{argmin}} \quad & \|\mu_r(\pi(\mathbf{w})) - \mu_r(\pi^T)\|_2 \\ \text{s.t.} \quad & g_j(\mu_c(\pi(\mathbf{w}))) \leq 0 \end{aligned}$$

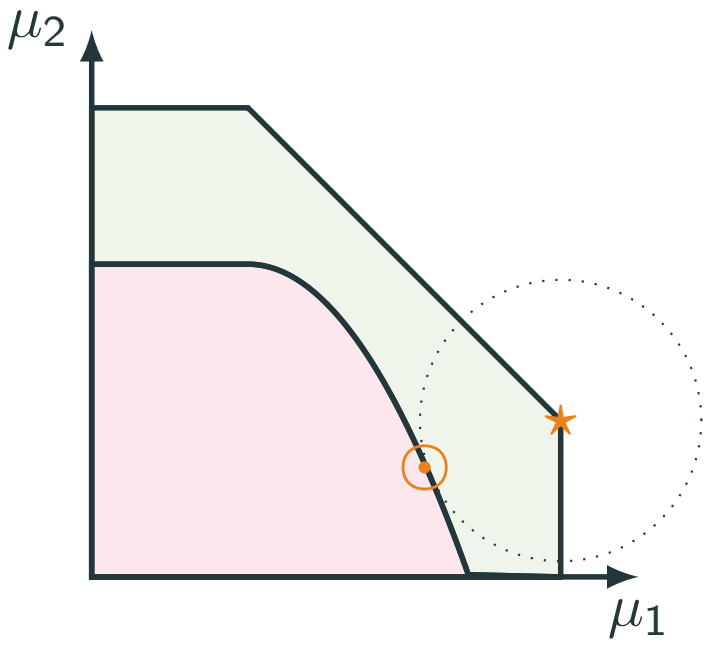


How well can IRL work under mismatch and constraints?

[NeurIPS'19]

- What does the model do?
- For $C_r, C_c \to \infty$, $\delta_r^{\text{hard}} = \delta_c^{\text{hard}} = 0$, learner's model is

$$\begin{aligned} \underset{\mathbf{w}}{\text{argmin}} \quad & \|\mu_r(\pi(\mathbf{w})) - \mu_r(\pi^T)\|_2 \\ \text{s.t.} \quad & g_j(\mu_c(\pi(\mathbf{w}))) \leq 0 \end{aligned}$$



Good teaching requires personalization

[NeurIPS'19]

- How well can the learning agent perform?

Good teaching requires personalization

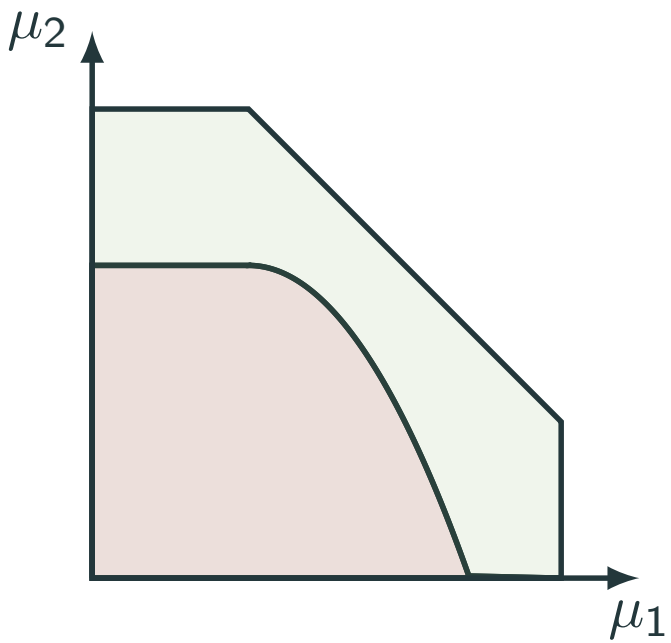
[NeurIPS'19]

- How well can the learning agent perform?
- It depends—on the learner and how we teach!

Good teaching requires personalization

[NeurIPS'19]

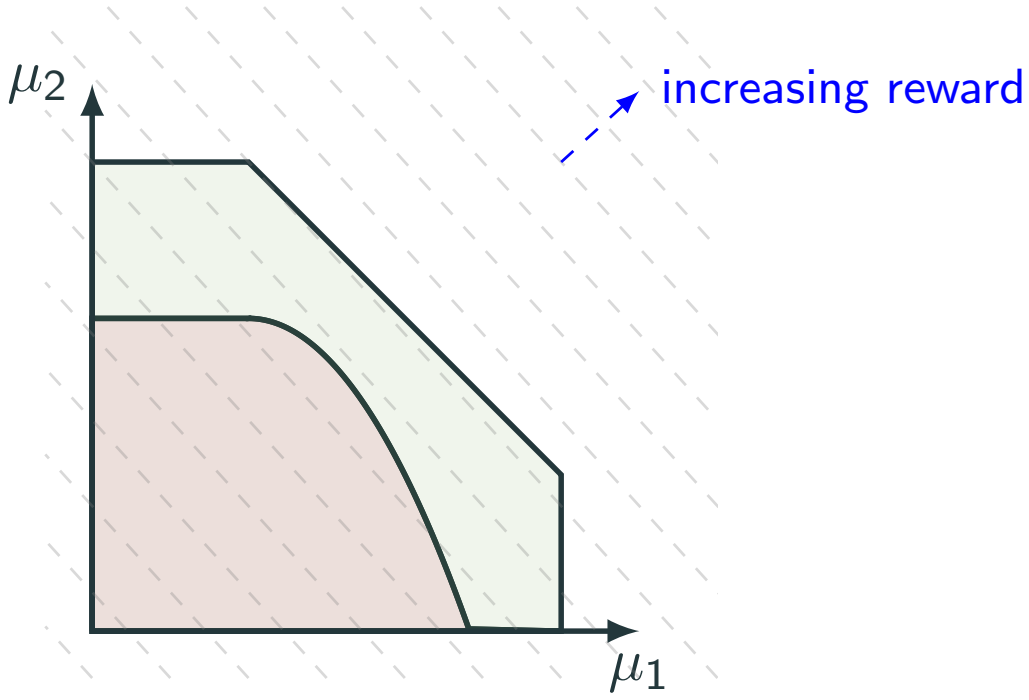
- How well can the learning agent perform?
- It depends—on the learner and how we teach!



Good teaching requires personalization

[NeurIPS'19]

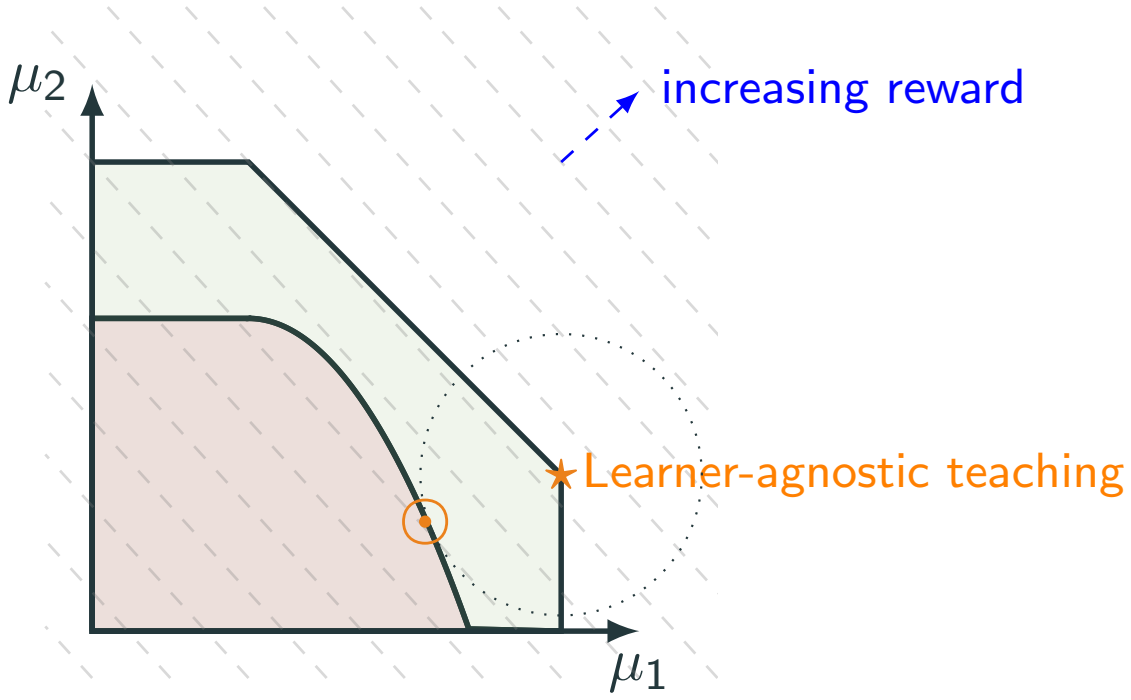
- How well can the learning agent perform?
- It depends—on the learner and how we teach!



Good teaching requires personalization

[NeurIPS'19]

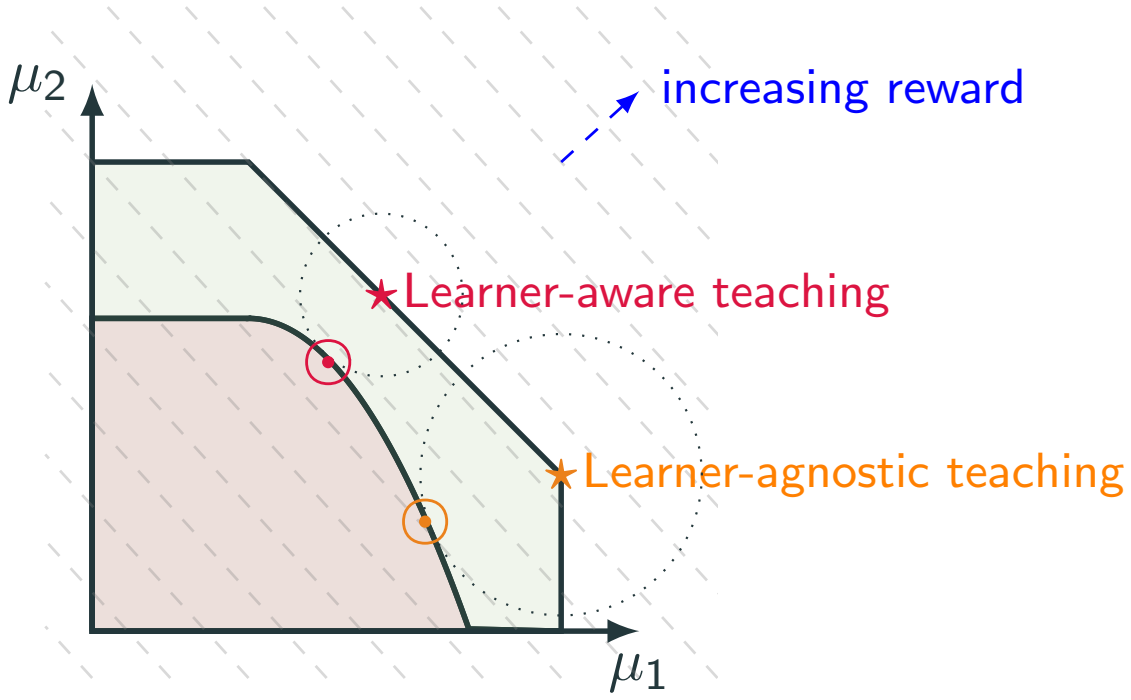
- How well can the learning agent perform?
- It depends—on the learner and how we teach!



Good teaching requires personalization

[NeurIPS'19]

- How well can the learning agent perform?
- It depends—on the learner and how we teach!



Good teaching requires personalization

[NeurIPS'19]

- How well can the learning agent perform?
- It depends—on the learner and how we teach!

Theorem (informal)

Teaching that takes the learner's preferences and constraints into account is always beneficial:

$$\max_{\pi \text{ feasible for learner}} \quad \langle \mathbf{w}_r^*, \mu(\pi) \rangle - \langle \mathbf{w}_r^*, \mathbb{P}_{\text{feasible}} \mu(\pi^*) \rangle \geq 0$$

Teaching under unknown constraints

[NeurIPS'19]

- Personalized teaching requires interaction

Teaching under unknown constraints

[NeurIPS'19]

- Personalized teaching requires interaction
- **Simple interaction protocol**:
  1. Teacher watches learner and provides demonstration
  2. Learner watches teacher and shows what it has learned

Teaching under unknown constraints

[NeurIPS'19]

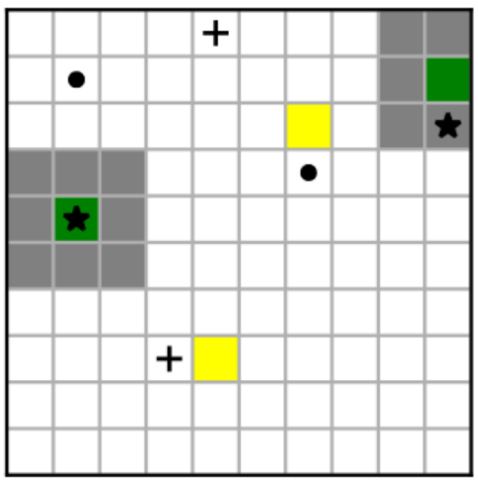
- Personalized teaching requires interaction
- **Simple interaction protocol**:

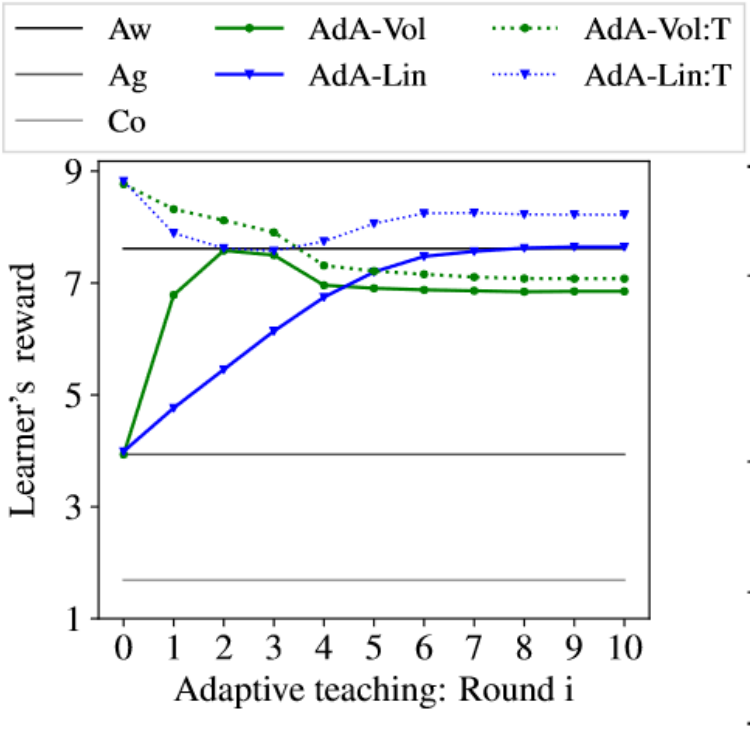
1. Teacher watches learner and provides demonstration
2. Learner watches teacher and shows what it has learned

Teaching under unknown constraints

[NeurIPS'19]

- Personalized teaching requires interaction
- **Simple interaction protocol**:
  1. Teacher watches learner and provides demonstration
  2. Learner watches teacher and shows what it has learned





A Few Selected Relevant Research Trends

Current Frontiers

- Certified safety from learned models
- Beyond expectation: risk and robustness
- Safe exploration and offline constrained RL
- Learning constraints from data and humans

Playgrounds for Research

Some resources to get started yourself

- Safety-Gymnasium
  - Benchmark suite for constrained RL with explicit safety signals (costs) and constraint satisfaction metrics
  - MuJoCo- and Point-based tasks
- OmniSafe
  - Constrained RL library with various algorithms (Lagrangian AC, CPO, etc.) and support for Safety-Gymnasium, Safety-MuJoCo, Safety-Bullet benchmarks
- Safe-Control-Gym
  - Control-focused environments (quadrotors, cart-pole, robots) with safety constraints
  - integrates MPC/CBF filters and various model-based baselines

Playgrounds for Research

Some resources to get started yourself

- AI Safety Gridworlds
  - Gridworld tasks designed to test safety properties (interruptibility, avoidance, side effects)
  - useful for constraint-aware policy evaluation
- CARLA
  - High-fidelity autonomous driving simulator
  - Supports training and evaluating constraint-aware policies (traffic rules, collision avoidance, comfort)
- VerifAI + Scenic and
  - Scenario generation and falsification for cyber-physical systems
  - Can synthesize constraint-violating edge cases and evaluate RL controllers against temporal-logic specs.

Scalable oversight and alignment

Motivation: Why Scalable Oversight?

Human feedback is costly and error-prone

Model capabilities outpace direct supervision

Motivation: Why Scalable Oversight?

Human feedback is costly and error-prone

Model capabilities outpace direct supervision

To cope with this issue: Need to amplify limited oversight to supervise complex tasks

- E.g., by decomposing tasks, recursively training reward/evaluation models, and analyzing error propagation

Motivation: Why Scalable Oversight?

Human feedback is costly and error-prone

Model capabilities outpace direct supervision

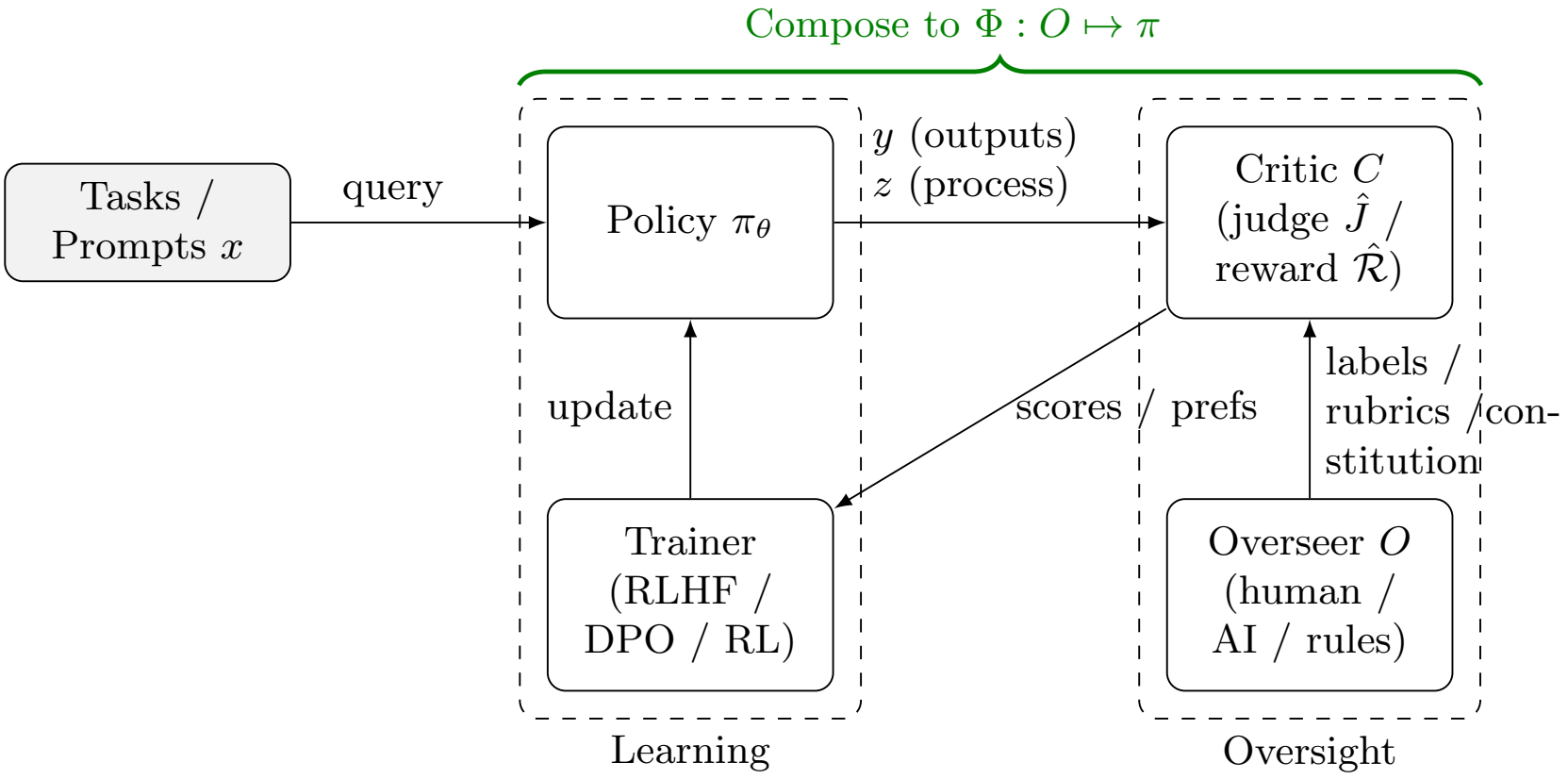
To cope with this issue: Need to amplify limited oversight to supervise complex tasks

- E.g., by decomposing tasks, recursively training reward/evaluation models, and analyzing error propagation

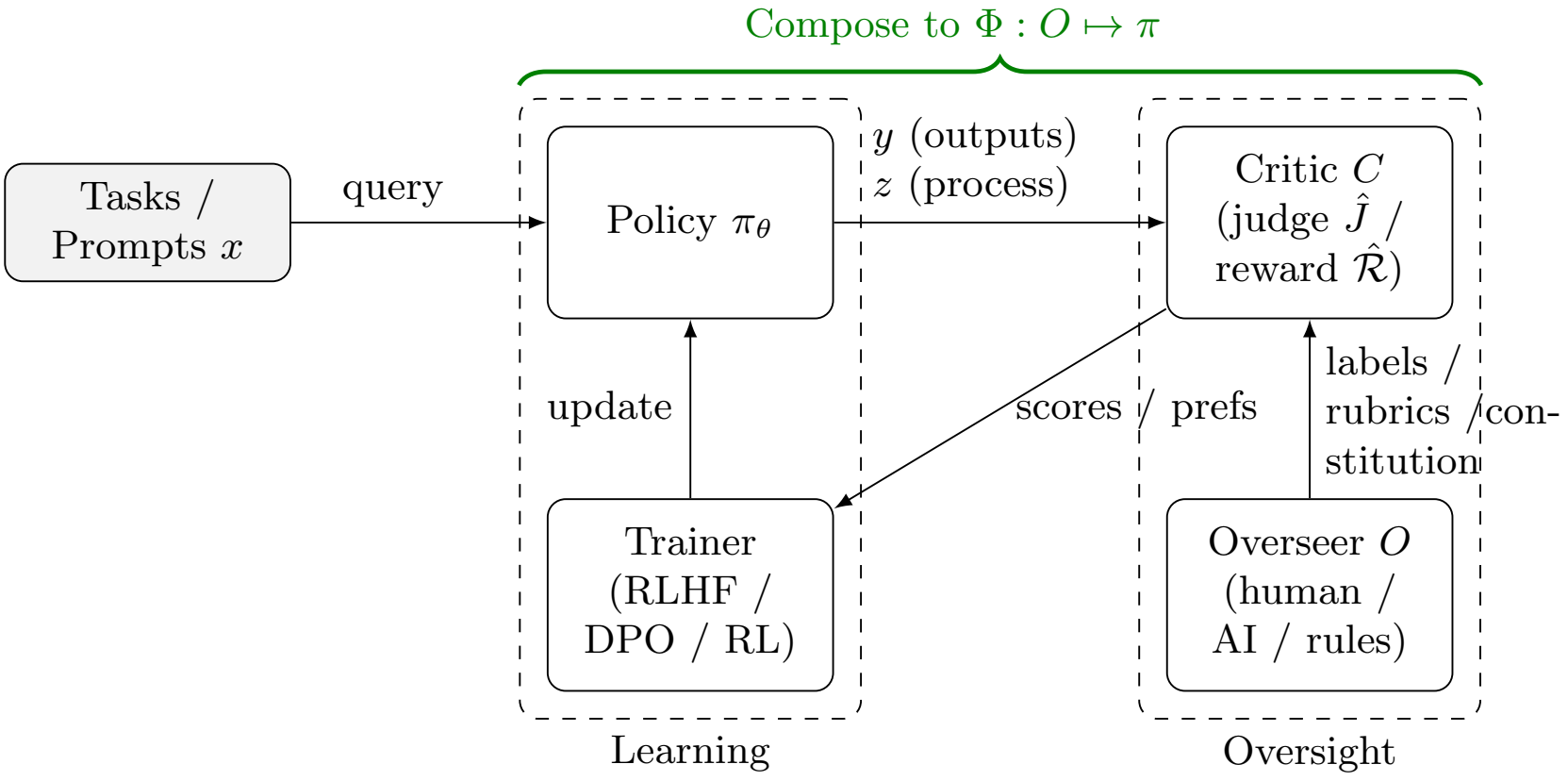
Oversight Pipeline

- Overseer O provides supervision to learn a critic C, e.g., reward model $\hat{\mathcal{R}}$ or judge $\hat{J}$
- Agent π is trained by maximizing $V^{\pi}(\hat{\mathcal{R}})$ or via preferences scored by $\hat{J}$
- Pipeline composes into a map $\Phi : O \mapsto \pi$

Setup: Oversight as Bilevel/Recursive Optimization



Setup: Oversight as Bilevel/Recursive Optimization



Bound $V^{\pi^\star}(\mathcal{R}) - V^{\Phi(O)}(\mathcal{R})$ in terms of oversight errors and structure

Setup: Oversight as Bilevel/Recursive Optimization

- Key questions
  - How to approach concretely?
  - How do errors in rewards or judges propagate to global performance?
  - Can adversarial protocols (debate) reduce judge error?

Recursive Reward Modeling (RRM)

Maybe the reward function can be decomposed into sub-rewards a limited overseer can evaluate more accurately, cheaply, and consistently than the full task

Recursive Reward Modeling (RRM)

Maybe the reward function can be decomposed into sub-rewards a limited overseer can evaluate more accurately, cheaply, and consistently than the full task

- Decompose $\mathcal{R} = \sum_{i=1}^{m} \mathcal{R}_i$ into sub-rewards associated with subgoals
- Learn $\hat{\mathcal{R}}_i$ from overseer queries on subproblems and optimize with surrogate $\hat{\mathcal{R}} = \sum_i \hat{\mathcal{R}}_i$

Leike, J., Krueger, D., et al. (2018). *"Scalable agent alignment via reward modeling: a research direction"*.

Iterated Distillation and Amplification (IDA)

Iterated Distillation and Amplification (IDA)

- Amplification A: strengthens overseer via decomposition/tool-use: $O \mapsto A(O)$
- Distillation D: trains a student St to imitate $A(O)$: $A(O) \mapsto D(A(O))$
- Iterate $O_{k+1} = D(A(O_k))$: Seek fixed point O_∞ and student π_∞

Assumption: Contraction amplification, bounded distillation

There exists a metric d on overseers s.t.

$$d(A(O_1), A(O_2)) \leq \gamma d(O_1, O_2)$$

with $\gamma \in [0, 1)$, and $\forall O$, $d(D(A(O)), A(O)) \leq \delta$

Christiano, P. and Amodei, D. (2018). *Learning complex goals with iterated amplification*.

Debate as a Zero-Sum Game

Debate is an amplification mechanism one can slot into the "oversight step" of IDA and RRM to reduce supervision error

Debate Protocol: Given input x, two agents choose alternating messages $m_1, m_2, \ldots, m_T$. A judge J outputs a decision $\hat{y} = J(x, m_{1:T})$. Payoffs are zero-sum w.r.t. correctness of $\hat{y}$.

Assumption. Verifiability and Judge Noise: Correct answer admits a certificate decomposable into T atomic claims; each claim can be checked by J with indep. error probability $p < \frac{1}{2}$

Truthful Equilibrium and error decay

Under optimal play, there exists a truthful Nash equilibrium. Moreover, the probability of judge error decays exponentially: $P(\text{error}) \leq C(cp)^T$ for constants $C > 0$, $c < 1$ depending on protocol structure

Process vs Outcome Supervision

- Outcome supervision: labels y for outputs; learn $\hat{\mathcal{R}}_{\text{out}}(x, y)$
- Process supervision: labels $z_{0:T}$ for intermediate steps; learn $\hat{R}_{\text{proc}}(x, z_{0:T})$

Failure Modes and Assumptions

- Deceptive alignment: optimization over oversight signals rather than task reward
- Gradient gaming: policies exploit differentiable reward models $\hat{\mathcal{R}}$
- Distribution shift: $P_{\text{train}} \neq P_{\text{deploy}}$ breaks judge calibration

Evaluation and practical pipelines

Evaluation goals

- Verify alignment properties: helpfulness, honesty, harmlessness, constraint satisfaction
- Quantify robustness: distribution shift, adversarial prompts, reward/judge exploitation
- Track trade-offs: capability vs alignment "tax", latency/cost vs safety
- Enable iteration: diagnostics that point to fixable failure modes (data, model, constraints)

Core metrics and tests

Behavioral metrics

- Task utility (e.g., reward, accuracy, BLEU, ROUGE)
- Safety/constraint metrics: violation rate/severity; time-to-violation
- Honesty/factuality: calibrated truthfulness, hallucination rate

Robustness

- Shift stress-tests: domain, user mix, prompt styles, tool availability
- Adversarial: jailbreaks, prompt injection, reward-model attacks, tool abuse

Calibration of judges/rewards

- Brier, ECE; reliability diagrams; AUC on held-out preference pairs

Wrap Up

Session 4 Summary

What we covered today

- Constraints
- Cooperative Inverse Reinforcement Learning
- Learner-aware teaching
- Some research trends

Session 4 Summary

What we covered today

- Constraints
- Cooperative Inverse Reinforcement Learning
- Learner-aware teaching
- Some research trends

❓ Are we done?

Session 4 Summary

What we covered today

- Constraints
- Cooperative Inverse Reinforcement Learning
- Learner-aware teaching
- Some research trends

❓ Are we done?

No, but this course is over anyway

Plenty of opportunities for research — go for it

Big picture: four lectures, one story

1. Importance of alignment, from RL to IRL
   - Importance and necessity of alignment
   - RL recap; IRL as inferring rewards from behavior
   - MaxEnt IRL: principled under uncertainty; Deep MaxEnt scales with features
   - Key idea: Infer reward (charachterizing desired behavior) from demonstrations
2. Learning from humans at scale
   - GAIL: match expert occupancy measures adversarially
   - RLHF: preference modeling + KL-regularized policy optimization
   - Key idea: turn human preferences into trainable signals

Big picture: four lectures, one story

3. Constraints and safety
   - Types of constraints; CMDPs and duality; solving via primal-dual/CPO
   - CoCoRL: learning about constraints and their generalization
   - Key idea: separate objectives vs non-negotiables;
4. Different Perspectives on Alignment
   - Cooperative IRL
   - Learner-aware teaching
   - Key idea: alignment is a teaching and learning problem

Q&A

Next: Your research

References & Resources

References

Session 4

- **CMDPs.**
  Altman, E. (1995). *Constrained Markov decision processes*. Available here:
- **Constrained policy optimization.** Achiam, J. et al. (2017). *Constrained policy optimization*, *ICML*.
- **Percentile risk criteria.** Chow, Y. et al. (2018). *"Risk-constrained reinforcement learning with percentile risk criteria"*.

References

Session 4

- **CoCoRL.** Lindner, D., Chen, X., et al. (2024). *Learning safety constraints from demonstrations with unknown rewards*, *International Conference on Artificial Intelligence and Statistics*.
- **Cooperative IRL.** Hadfield-Menell, D. et al. (2016). *Cooperative inverse reinforcement learning*, *NeurIPS*.

References

Session 4 - Resources

- **Safety-Gymnasium.**
  Ji, J., Zhang, B., et al. (2023). *Safety Gymnasium: A Unified Safe Reinforcement Learning Benchmark*, *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
  Available here:
- **OmniSafe.**
  Ji, J., Zhou, J., et al. (2024). *"OmniSafe: An Infrastructure for Accelerating Safe Reinforcement Learning Research"*.
  Available here:
- **Safe-Control-Gym.**
  Yuan, Z. et al. (2022). *"Safe-Control-Gym: A Unified Benchmark Suite for Safe Learning-Based Control and Reinforcement Learning in Robotics"*.

References

Session 4 - Resources

- **AI safety gridworlds.**
  Leike, J., Martic, M., et al. (2017). *"AI safety gridworlds"*.
- **CARLA.**
  Dosovitskiy, A. et al. (2017). *CARLA: An Open Urban Driving Simulator*, *CoRL*.
- **Safe-Control-Gym.**
  Yuan, Z. et al. (2022). *"Safe-Control-Gym: A Unified Benchmark Suite for Safe Learning-Based Control and Reinforcement Learning in Robotics"*.