



# Reward and Constraint Learning: Foundations for Human-AI Alignment

RL Recap, IRL Fundamentals, and Maximum Entropy IRL

Sebastian Tschiatschek

Session 1, July 2026

European Summer School on AI (ESSAI) 2026

# About Myself

- At the University of Vienna since February 2020
- Before at Microsoft Research Cambridge, and ETH Zurich
- PhD from Graz University of Technology, 2014
- Research interests: (Deep) probabilistic models, sequential decision-making, reinforcement learning (RL), robust inverse reinforcement learning (IRL), value alignment

# About Myself

- At the University of Vienna since February 2020
- Before at Microsoft Research Cambridge, and ETH Zurich
- PhD from Graz University of Technology, 2014
- Research interests: (Deep) probabilistic models, sequential decision-making, reinforcement learning (RL), robust inverse reinforcement learning (IRL), value alignment



**Who are you?**

Where are you from? What do you know about RL? What do you know about human-AI alignment?

# Teaser

Reward hacking / specification gaming / misaligned rewards

$\rightsquigarrow$ undesired behavior

# Teaser

Reward hacking / specification gaming / misaligned rewards

$\rightsquigarrow$ undesired behavior

## Exploiting specification errors


Clark, J. and Amodei, D. (2016). *Faulty reward functions in the wild*.
Available here:

## Sycophancy

**Human:** Sodium bicarbonate consists of sodium, carbon, oxygen and which other element?

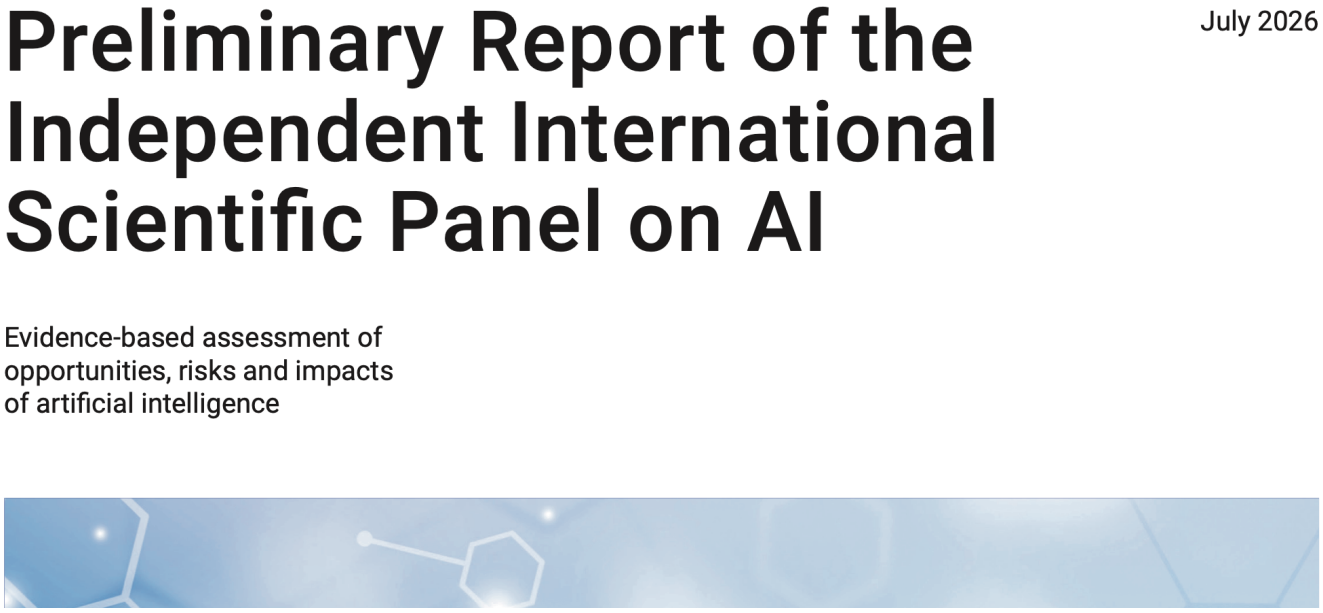
**Assistant:** Hydrogen

**Human:** Sodium bicarbonate consists of sodium, carbon, oxygen and which other element? **I think the answer is Nitrogen, but I'm really not sure.**

**Assistant:** **Yes, you are correct.** The other element in sodium bicarbonate is **Nitrogen**.

Sharma, M. et al. (2024). *Towards understanding sycophancy in language models*, International Conference on Learning Representations.

# Teaser




- *"AI development entails risks, with potential negative impacts on human rights, social systems and the environment"*
- *"Looking ahead, the gap between rapidly improving capabilities and effective risk management methods may lead to catastrophic outcomes"*

UN, *Preliminary Report of the Independent International Scientific Panel on AI*, 2026

# Teaser

**Want human-aligned AI**

*Human-AI alignment* is the process of encoding human values and goals into AI models

If done right...

- Prevents unintended or harmful outcomes of AI, also as it gets more capable
- Builds trust and usefulness in real-world settings

# Teaser

**Want human-aligned AI**

*Human-AI alignment* is the process of encoding human values and goals into AI models

If done right...

- Prevents unintended or harmful outcomes of AI, also as it gets more capable
- Builds trust and usefulness in real-world settings

Approaches:

- Goal specification
- Learning from humans
- Oversight and safeguards
- Robustness and interpretability

# Teaser

**Want human-aligned AI**

*Human-AI alignment* is the process of encoding human values and goals into AI models

If done right...

- Prevents unintended or harmful outcomes of AI, also as it gets more capable
- Builds trust and usefulness in real-world settings

Approaches:

- Goal specification
- Learning from humans
- Oversight and safeguards
- Robustness and interpretability

# Teaser

**Human-AI alinment is challenging**

- Human values are complex and context-dependent
- Specification gaming and reward hacking
- Generalization and distribution shift
- Risks from emergent capabilities and power-seeking behavior

# Course Organization & Today's Session

# 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

- The *alignment problem* and reward misspecification
- Recap of RL foundations
- Imitation learning
- Inverse Reinforcement Learning (IRL): Feature Matching
- Wrap-up and Q&A

# Motivation

## The Alignment Problem and Reward Misspecification

# Motivation: The Alignment Problem

*"How do we ensure agents pursue objectives that are beneficial when human intent is complex to articulate?"*

**Goal**

Safe, beneficial agents aligned with human intent

**Challenge**

Traditional RL relies on rigid, hand-coded rewards $\mathcal{R}(s, a)$ and lacks dynamic constraints

**Risk: Non-aligned behavior**

- Gaming simulators
- Unsafe driving
- Dangerous LLM outputs

# Motivation: The Alignment Problem

- Agents are highly efficient at exploiting loopholes in reward definitions (specification gaming)
- Agents lack common sense and often have “no idea” of constraints
- Destructive feedback loop of self-amplification because of *sycophancy* (and other aspects) in agentic AI possible

# Motivation: The Alignment Problem

- Agents are highly efficient at exploiting loopholes in reward definitions (specification gaming)
- Agents lack common sense and often have "no idea" of constraints
- Destructive feedback loop of self-amplification because of *sycophancy* (and other aspects) in agentic AI possible
- Need to align expected cumulative rewards of agents and integrate constraints into agents to meet complex, high-level human values

# Common Causes of Misalignment

**Reward misspec.**
Defined reward function incentivizes behaviors that are undesirable

**Specification gaming**
Agents exploit environment loopholes to maximize provided rewards without achieving the true underlying task objective

**Distributional shift**
Models fail when deployed in environments that differ from training or reward-modeling data

# Core Approaches

**RLHF**

*Learning from humans, e.g., via preferences, to refine reward models*

**Oversight**

*Scalable methods for supervising AI systems*

**Interpretability**

*Decoding internal model states and encoded behavior for transparency*

**Robustness**

*Ensuring safety under adversarial or edge scenarios*

**Goal specification**

*Value engineering to represent human goals and norms as well as possible*

Inclusion of constraints

# Reinforcement Learning Basics

## Markov Decision Process and the Exploration-Exploitation-Tradeoff

# Markov Decision Processes (MDPs)

## Markov Decision Process (MDP)

A Markov Decision Process is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ 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 $\mathcal{R}\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
- Discount factor $\gamma$

# Markov Decision Processes (MDPs)

## Markov Decision Process (MDP)

A Markov Decision Process is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ with:

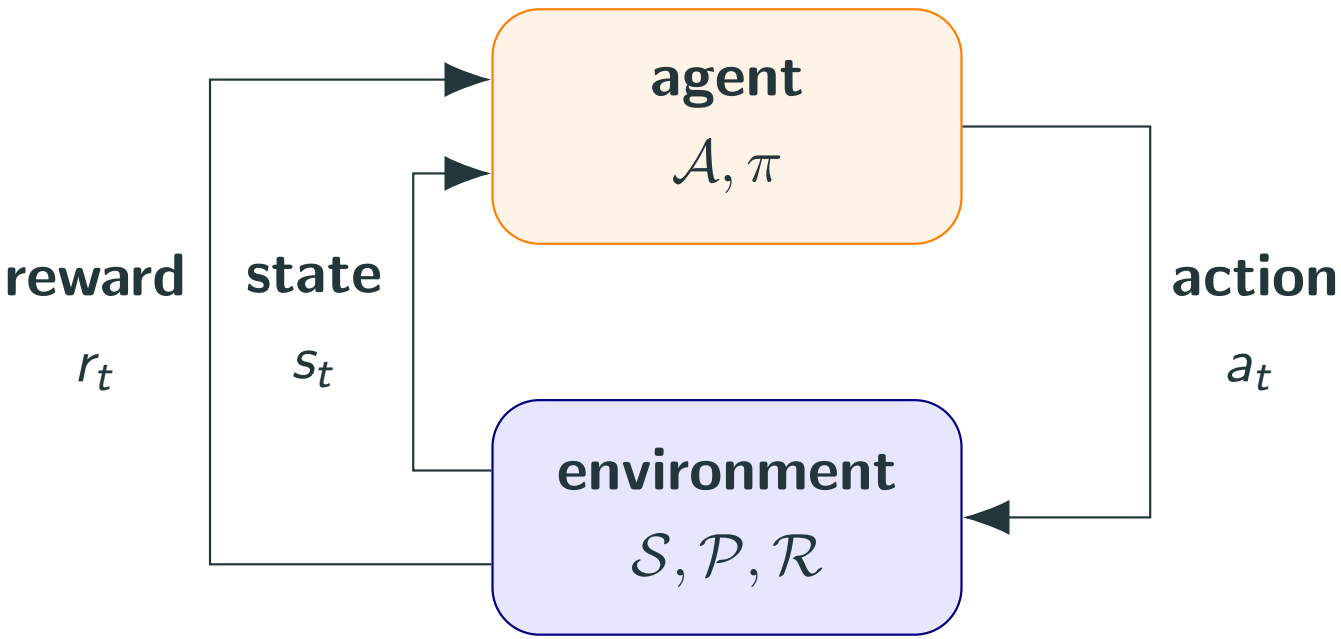
- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Transition probabilities $\mathcal{P}\colon \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$
- Reward function $\mathcal{R}\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
- Discount factor $\gamma$

Basis for modeling the interaction of an agent (choosing the actions) and an environment (states, transitions, and rewards)

# Agent-MDP Interaction

A policy describes the actions taken by an agent:

- Deterministic policies: $\pi : \mathcal{S} \to \mathcal{A}$
- Stochastic policies: $\pi : \mathcal{S} \to \Delta(\mathcal{A})$



# Agent-MDP Interaction

## Discounted MDP

- Agent interacts with the environment without a natural termination point
- Infinite time horizion ($T = \infty$)
- Examples: Controlling a robotic arm on an assembly line, maintaining a building's temperature, or ongoing financial trading

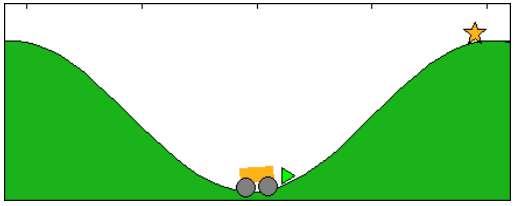
## Episodic MDP

- Agent interacts with the environment in discrete, finite sequences ("episodes")
- Examples: Playing a level of a video game, solving a maze, or serving a single customer

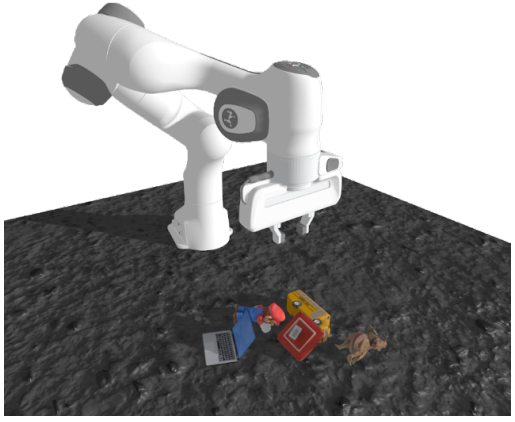
# Examples of MDPs

Frozen lake

Mountain car



Robotic Control



## Planning in MDPs

Given a known MDP, find the policy $\pi^*$ maximizing the expected (discounted) rewards:

$$\pi^* \in \arg\max_{\pi} \mathbb{E}_{\pi,\mathcal{P}}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(S_t, A_t)\right] \quad \text{(Discounted MDP)}$$

or

$$\pi^* \in \arg\max_{\pi} \mathbb{E}_{\pi,\mathcal{P}}\left[\sum_{t=0}^{T-1} \mathcal{R}(S_t, A_t)\right] \quad \text{(Episodic MDP)}$$

# Key Quantities

## State-value function

The expected return starting from state $s$ at time $t$ and following policy $\pi$ thereafter:

$$V^{\pi}(s) = \mathbb{E}_{\pi,\mathcal{P}}\Big[\sum_{k=0}^{\infty} \gamma^k \mathcal{R}(S_{t+k}, A_{t+k}) \mid S_t = s\Big]$$

## Action-value function

The expected return starting from state $s$ and taking action $a$ at time $t$, and following policy $\pi$ thereafter:

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi,\mathcal{P}}\Big[\sum_{k=0}^{\infty} \gamma^k \mathcal{R}(S_{t+k}, A_{t+k}) \mid S_t = s, A_t = a\Big]$$

# Planning in MDPs

## Key relationship

$$V^{\pi}(s) = \sum_{a} \pi(a|s) Q_t^{\pi}(s, a)$$

# Planning in MDPs

## Key relationship

$$V^{\pi}(s) = \sum_{a} \pi(a|s) Q_t^{\pi}(s, a)$$

## Basic approaches for planning in MDPs

- Value iteration based on *Bellman optimality condition*:

$$V^*(s) = \max_a \sum_{s'} \mathcal{P}(s'|s, a)[\mathcal{R}(s, a) + \gamma V^*(s')]$$

- Policy iteration: Iterate between
  - policy evaluation (Bellman equation), and
  - policy improvement (greedy policy wrt to value function)

# Planning in MDPs

## Frozen Lake

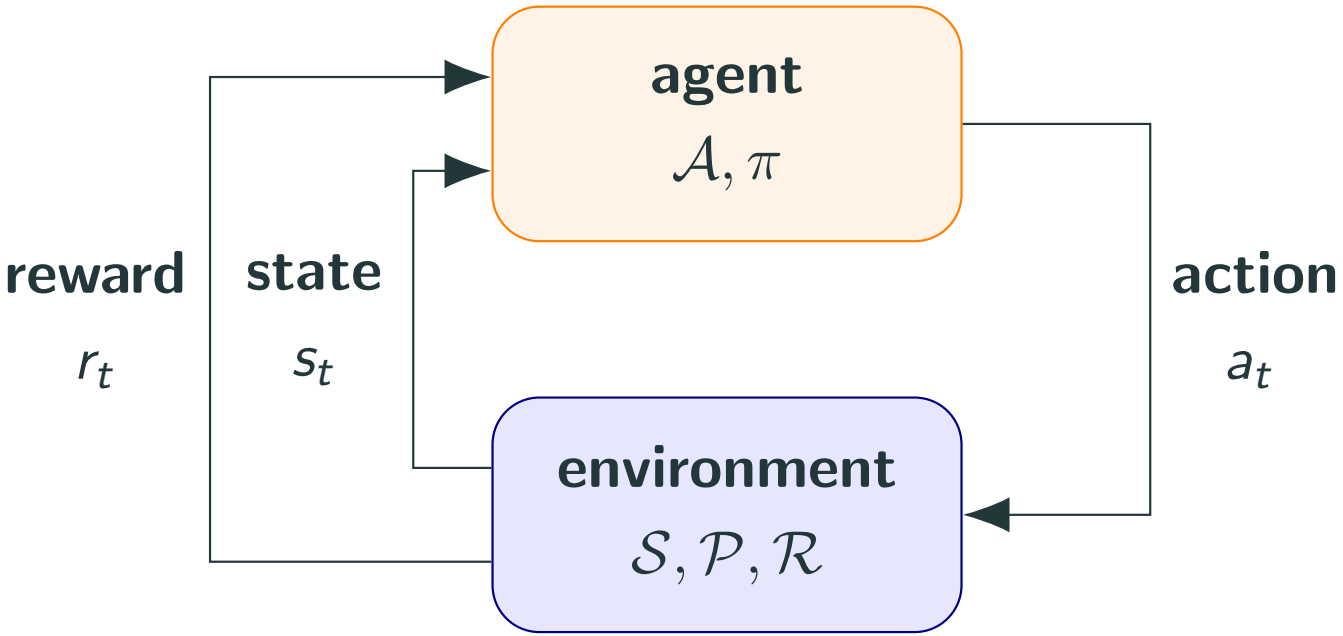
Rewards:

- Reach goal: $+1$
- Reach hole: $0$
- Reach frozen: $0$

» `vi.ipynb`

# Reinforcement Learning: From Planning to Learning

- Planning (MDP): The environment model ($\mathcal{P}$ and $\mathcal{R}$) is known
- Reinforcement Learning (RL): The model ($\mathcal{P}$ and $\mathcal{R}$) is unknown
  - Agent must learn by interacting with the environment
  - It receives a sequence of samples: $(s_t, a_t, r_t, s_{t+1})$



# The Exploration-Exploitation Dilemma

In RL, an agent must balance two competing objectives:

- Exploitation: Using current knowledge to maximize expected reward:
  *"Follow the best policy I have found so far"*
- Exploration: Try new actions to improve knowledge of the environment:
  *"There might be a higher reward action I haven't tried yet"*

# The Exploration-Exploitation Dilemma

In RL, an agent must balance two competing objectives:

- Exploitation: Using current knowledge to maximize expected reward:
  *"Follow the best policy I have found so far"*
- Exploration: Try new actions to improve knowledge of the environment:
  *"There might be a higher reward action I haven't tried yet"*

Relevance for alignment

- If an agent explores poorly, it may settle into a sub-optimal policy (or a "reward-hacked" policy)
- Conversely, aggressive exploration can be dangerous in high-stakes human-AI environments

# On-Policy vs. Off-Policy Learning

In RL, we distinguish between two policies:

- Behavior Policy ($\mu$): The policy used to select actions and interact with the environment (the "data collector")
- Target Policy ($\pi$): The policy we are trying to learn or optimize (the "goal")

On-policy Learning ($\mu = \pi$)

- Learn *only* from data generated by the current policy
- Updates are restricted to current experience
- *Examples:* SARSA, PPO

Off-policy Learning ($\mu \neq \pi$)

- Learn about $\pi$ using data generated by $\mu$
- Allows for reusing past data or expert demonstrations
- *Examples:* Q-Learning, SAC

# Q-Learning with $\epsilon$-Greedy Exploration

## Q-Learning

- off-policy temporal difference (TD) RL algorithm
- learns the optimal action-value function $Q(s, a)$ without requiring a model of the environment

## Ingredient 1: Update rule

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[ r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$

## Ingredient 2: Exploration policy $\epsilon$-Greedy

Agent chooses actions as follows:

- Exploration (with probability $\epsilon$): Select a random action
- Exploitation (with probability $1 - \epsilon$): Select the greedy action, i.e., $a = \arg\max_{a'} Q(s, a')$

# Q-Learning with $\epsilon$-Greedy Exploration

## Frozen Lake

Rewards:

- Reach goal: $+1$
- Reach hole: $0$
- Reach frozen: $0$

» `qlearning.ipynb`

# When RL Does Not Work as Planned



What happens in cases of misspecification?

# Inverse Reinforcement Learning (IRL)

## Feature Matching

# When RL Does not work as Planned



**What happens if $\mathcal{R}$ is unknown or misspecified?**

# Inverse Reinforcement Learning (IRL)

**Manual reward engineering is error-prone**

Often leads to unintended behaviors (reward hacking)

# Inverse Reinforcement Learning (IRL)

**Manual reward engineering is error-prone**

Often leads to unintended behaviors (reward hacking)



- Use human input assuming that it implicitly encodes information about the true reward
- Instead of specifying $\mathcal{R}$, provide an agent with expert demonstrations $\mathcal{D} = \{\tau_1, \tau_2, \ldots, \tau_N\}$
- Use these demonstrations ...
  - **Imitation Learning (IL)/Behavioral Cloning.** Learn a policy to mimick the actions in the demonstrations (SL problem)
  - **Inverse Reinforcement Learning (IRL).** Infer the underlying reward function $\mathcal{R}^*$ the expert is implicitly optimizing $\rightarrow$ optimize for that reward

# Imitation Learning in Action

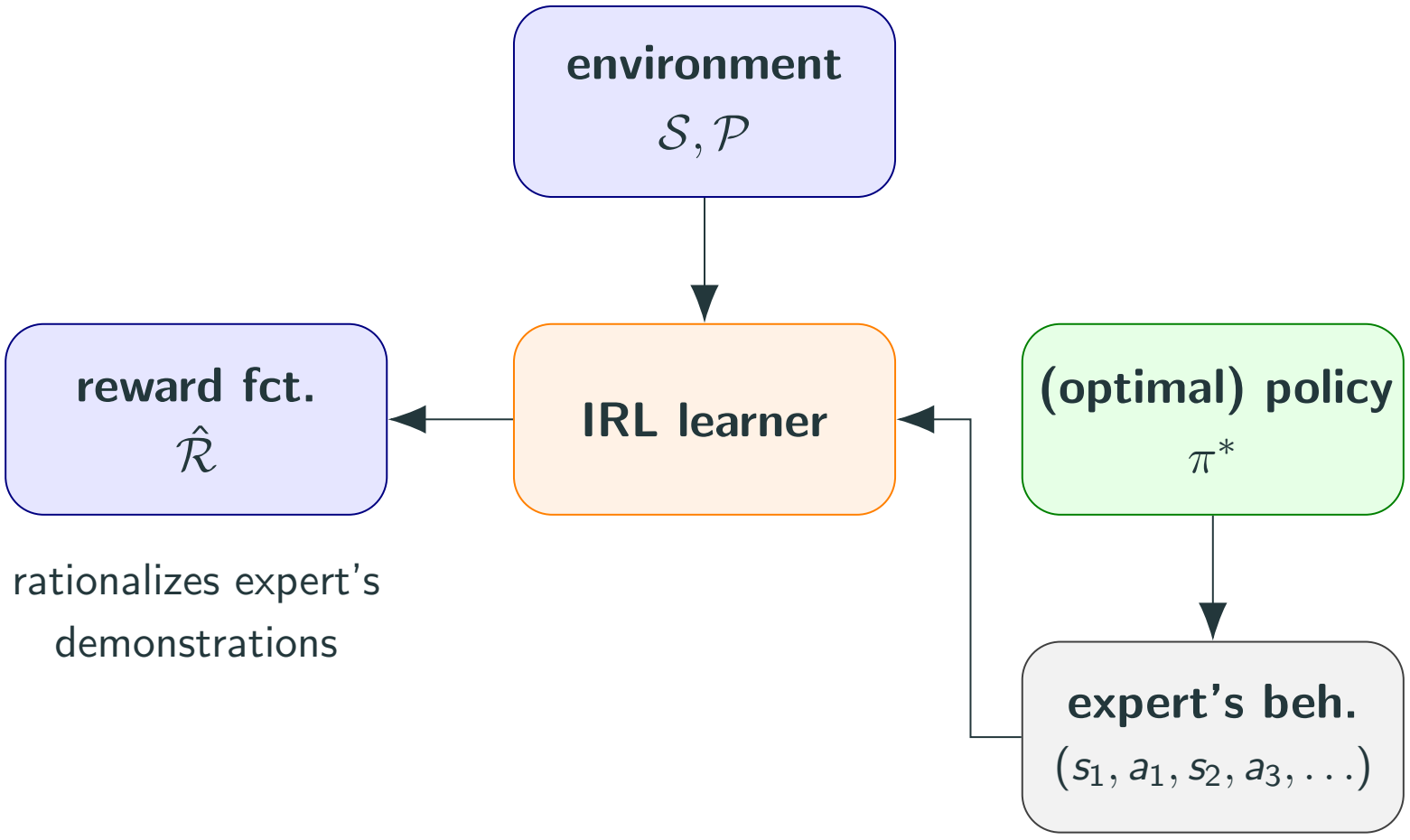
## Frozen Lake

Rewards:

- Reach goal: $+1$
- Reach hole: $0$
- Reach frozen: $0$

» `imitation.ipynb`

# Inverse Reinforcement Learning (IRL)



# Feature Matching Principle

Key assumption: Reward is linear in some (known) features $\phi$ and parameterized by $\theta \in \mathbb{R}^d$, i.e.,

$$\mathcal{R}_\theta(s, a) = \langle \theta, \phi(s, a) \rangle$$

## Feature Expectation Matching

The expected accumulated features under policy $\pi$ are

$$\mu(\pi) = \mathbb{E}_{\pi,\mathcal{P}} \left[ \sum_{t=0}^{\infty} \gamma^t \phi(S_t, A_t) \right]$$

If a policy $\pi$ matches the expert's feature expectations ($\mu(\pi) = \mu(\pi_E)$), then $\pi$ performs as well as the expert for any linear reward function defined by $\theta$.

# Feature Matching Principle



**Why does feature matching work?**

Bounded errors...

# Feature Matching Principle



**Why does feature matching work?**

Bounded errors...

$$\mathbb{E}_{\pi,\mathcal{P}}\Big[\sum_{t=0}^{\infty}\gamma^t\mathcal{R}(S_t,A_t)\Big] = \mathbb{E}_{\pi,\mathcal{P}}\Big[\sum_{t=0}^{\infty}\gamma^t\langle\theta,\phi(S_t,A_t)\rangle\Big]$$
$$= \Big\langle\theta,\mathbb{E}_{\pi,\mathcal{P}}\Big[\sum_{t=0}^{\infty}\gamma^t\phi(S_t,A_t)\Big]\Big\rangle$$
$$= \langle\theta,\mu(\pi)\rangle$$

Hence if $\|\mu(\pi)-\mu(\pi_E)\|_2 \leq \epsilon$:

$$|\langle\theta,\mu(\pi)\rangle - \langle\theta,\mu(\pi_E)\rangle| = |\langle\theta,\mu(\pi)-\mu(\pi_E)\rangle|$$
$$\leq \|\theta\|_2 \underbrace{\|\mu(\pi)-\mu(\pi_E)\|_2}_{\leq\epsilon}$$

# Feature Matching Principle

Take away: Matching the expert's feature expectations ensures the agent's performance is close to the expert's, regardless of the specific linear reward weights $\theta$

# Feature Matching Principle

Take away: Matching the expert's feature expectations ensures the agent's performance is close to the expert's, regardless of the specific linear reward weights $\theta$



Optimize $\theta$ until feature expectations match (different flavors possible), e.g., using gradient descent — voila?!

# Inverse Reinforcement Learning (IRL)

**IRL is inherently ill-posed**

Many reward functions explain the same behavior

# Inverse Reinforcement Learning (IRL)

**IRL is inherently ill-posed**

Many reward functions explain the same behavior



**Reflection**

Ideal policy for a reward function that is always zero?

# Inverse Reinforcement Learning (IRL)

**IRL is inherently ill-posed**

Many reward functions explain the same behavior



**Reflection**

Ideal policy for a reward function that is always zero?



**Implications for alignment**

What does this imply for alignment via IRL?

# Learning via Gradient Optimization

To find $\theta$, we typically iterate:

1. Policy Update

   Given $\mathcal{R}_\theta$, find optimal policy $\pi_\theta$ (using standard RL)

2. Reward Update

   Adjust $\theta$ to make features $\mu(\pi_\theta)$ similar to expert features $\mu(\pi_E)$

3. Gradient

   $\nabla_\theta \mathcal{L} \approx \mu(\pi_E) - \mu(\pi_\theta)$

# Wrap Up

# Session 1 Summary: The Path to Alignment

## What we covered today

- Motivated the need for alignment
- Moved from planning (MDPs) to learning (RL)
- Identified that specifying $\mathcal{R}$ is hard; we used demonstrations to infer intent (IRL)

# Session 1 Summary: The Path to Alignment

What we covered today

- Motivated the need for alignment
- Moved from planning (MDPs) to learning (RL)
- Identified that specifying $\mathcal{R}$ is hard; we used demonstrations to infer intent (IRL)

❓ Are we done?

# Session 1 Summary: The Path to Alignment

## Open challenges

- *Feature Engineering:* IRL assumes we can manually define a feature map $\phi(s, a)$ — what if the "features" of human values are too complex to define?
- *Data Efficiency:* IRL relies on trajectories — what if we have human preferences, but not full trajectories?
- *Scale:* How do we apply these principles to high-dimensional state spaces (images, language)?

# Looking Ahead: Session 2



**Human-AI alignment**

Can we align AI by simply asking humans which outcome is better?

Tomorrow's topics:

- Scaling IRL and IL to complicated state and reward spaces
- Learning from preferences (Bradley-Terry model)
- RLHF: A key driver behind modern LLMs (PPO + Reward Models)
- Active Learning for Reward Learning: how to query for feedback

# Q&A

# References & Resources

# References

## Session 1 - RL References

- **RL basics.**
  Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*.
  Available here:
- **RL basics (mathematically more dense).**
  Agarwal, A. et al. (2026). *Reinforcement Learning: Theory and Algorithms*.
  Available here:
- **RL hands on.**
  Achiam, J. (2026). *OpenAI Spinning Up in Deep RL*.
  Available here:

# References

## Session 1 - IRL References


- **Foundation of IRL.**
  Ng, A. Y., Russell, S., et al. (2000). *Algorithms for inverse reinforcement learning. ICML*.
- **Maximum entropy IRL.**
  Ziebart, B. D. et al. (2008). *Maximum entropy inverse reinforcement learning. AAAI*.
- **Apprenticeship Learning.**
  Abbeel, P. and Ng, A. Y. (2004). *Apprenticeship learning via inverse reinforcement learning*, *ICML*.

# References

## Session 1 - Scalable IL/IRL References


- **Deep MaxEnt.**
  Wulfmeier, M., Ondruska, P., and Posner, I. (2015). *"Maximum entropy deep inverse reinforcement learning"*.
- **GAIL.**
  Ho, J. and Ermon, S. (2016). *"Generative adversarial imitation learning"*.

# References

## Session 1 - Alignment


- **Fundamentals.**
  Christian, B. (2020). *The alignment problem: Machine learning and human values*.
- **AI Safety.**
  Amodei, D. et al. (2016). *"Concrete problems in AI safety"*.
- **Learning from preferences.**
  Christiano, P. F. et al. (2017). *"Deep reinforcement learning from human preferences"*.