

Reinforcement Learning in the Agentic Age

Andrew Perrault
perrault.17@osu.edu
CSE 5539

The big question

How do we build artificial **intelligence**?

What is intelligence?

Intelligence = capability (**task**-specific)

If a system can perform **task x** well, it is **intelligent** in **task x**

Intelligence \neq not a single quantity, though we might call a system “**generally intelligent**” if it can perform a lot of **tasks** well (which ones?)

Human intelligence

We define **intelligence** as the capabilities we see in humans, and the similar patterns we see in animals

Human **intelligence** has multiple dimensions

- We know this because we see humans who are very good at some things while be very bad at others

Savants and brain damage

Kim Peek (savant, inspiration for *Rain Man*): struggled to button a shirt, low IQ, could memorize 98% of a book in an hour, memorized 12,000 books

Phineas Gage: after brain damage, lost business/planning skill, but kept logic, reasoning and speech

Louis Victor Leborgne: after brain damage, lost speaking ability, but could still understand speech and do math

→ Strong evidence that human **intelligence** consists of many systems working together

Why are some people good at chess?

The most studied question in cognitive science

Chess ability is mostly the result of practice, which leads to the development of chess-specific **pattern recognition**

- Mild role for IQ, mostly at the beginner stage

Experiment: Polgar sisters

- Three sisters, raised to play chess from childhood
- Result: three world-class players, including the strongest female chess player of all time

→ **Task-specific intelligence** in humans is largely the result of practice, not **general intelligence**



Does chess ability **generalize** in humans?

Does learning chess in school improve math skills, reading comprehension or **general intelligence**?

- No (Sala and Gobet, 2017, etc.)

The “myth of far transfer”: getting good at difficult **tasks** (puzzles, chess, Latin) will make **not** you more **generally intelligent**

- Problem: improving at a **task** builds **task-specific pattern recognition** that only transfers to similar **tasks** (e.g., French → Spanish)

Does chess ability generalize in machines?

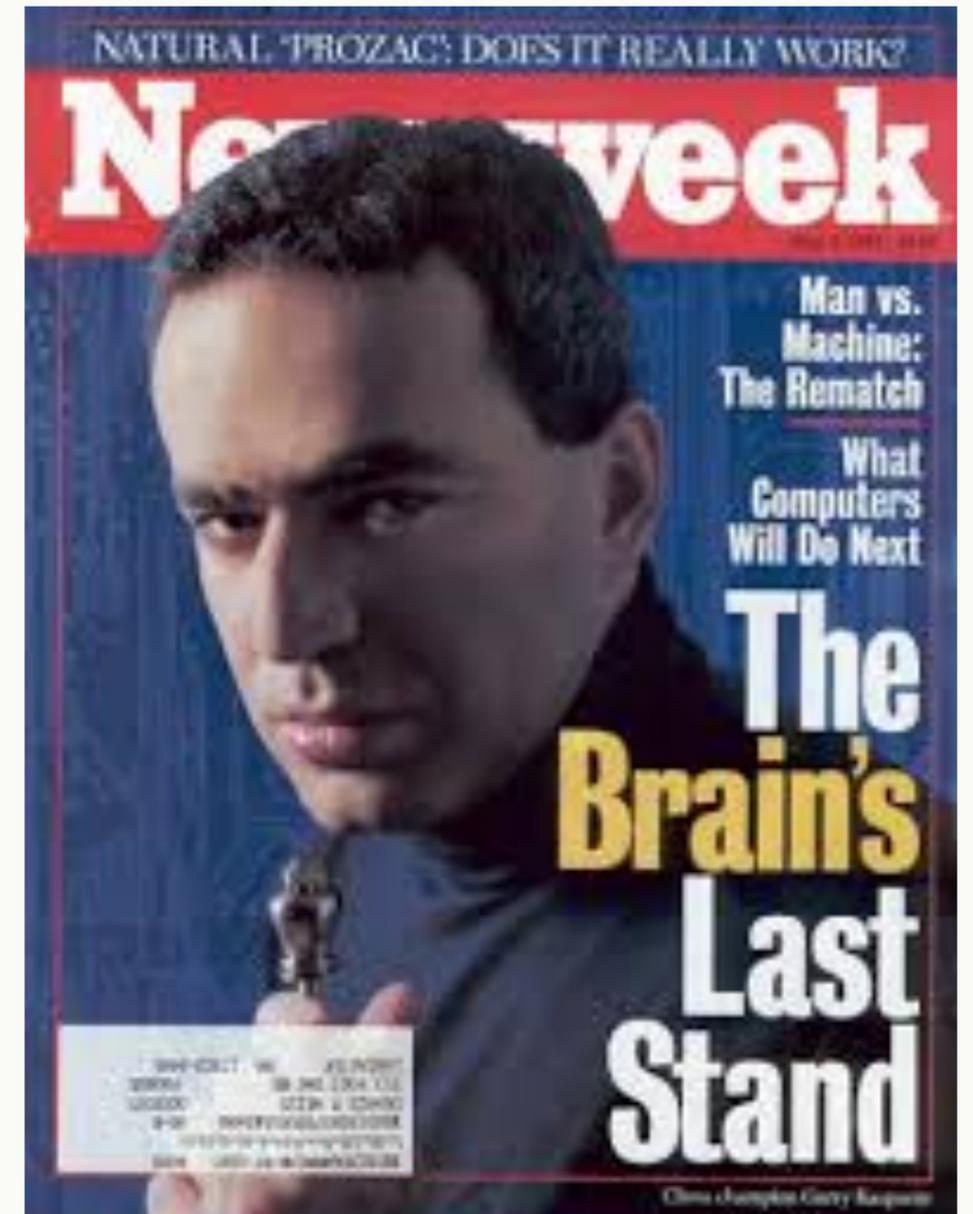
What do you think?

In general, no. We can build a system that is superhuman at chess (Deep Blue, 1990s) but can't do any other task. 😞

Chomsky 😡: "A computer beating a Grandmaster at chess is about as interesting as a bulldozer winning an Olympic weightlifting competition"

- "This position looks dangerous; I need to fortify the center." vs. "I see 200,000,000 outcomes. Outcome #49281 leads to a +0.03 advantage."

McCarthy 😡: "Computer chess has developed much as genetics might have if the geneticists had concentrated their efforts starting in 1910 on breeding racing *Drosophila* [fruit flies]. We would have some science, but mainly we would have very fast fruit flies."



My opinion

I understand why people were disappointed by DeepBlue/AlphaGo/Watson etc.

But it makes sense for a system with a computer brain to take advantage of computation

- If humans could evaluate chess positions really fast, human experts would do a lot more exhaustive search!

General biases:

- We want AI to do things in a human way, even if it doesn't make sense
- We value when AI systems do things that are hard for humans (even if they turn out to be easy for computers)
- We undervalue the ability of AI systems to do things that are “easy” for humans (e.g., locomotion, manipulation, keeping track of reality, not being gullible, see spatial relationships in images, detect emotion in speech)—these skills are needed because compound **tasks** require them

Narrow vs. general AI

Both narrow and general AI are useful

Narrow: AlphaFold (Hassabis and Jumper, 2024 Nobel), predicts protein structures. It is transforming structural biology and drug discovery—we effectively know the structure of every known protein (and this could \$100k per protein previously). But it must take a protein (or protein + X pair) as input and predict the outcome as output, can't do any other **task**

General: ChatGPT, etc. Can receive text, images (and sometimes audio) as input and produce text as output. This encompasses almost any problem. However, really bad at both protein structure prediction and chess (relative to our best systems).

How do we build artificial **intelligence**?

Two big ideas

Idea #1. **Imitate **intelligent** behaviors.** We have **intelligent** behaviors for lots of **tasks**. Let's build a system that reproduces those behaviors.

Idea #2. **Learning from experience.** Let's build environments that test a particular **task**. Then, systems can master those **tasks** through experimentation in the environment.

Intelligence via imitation

We have lots of **intelligent task** behavior data (e.g., from the internet), and we have figured out really clever ways to train machines to imitate those behaviors (e.g., with transformers)

We will think of **task** behavior as a sequence of states and actions
 $(s_0, a_0, \dots, s_t, a_t)$

- Training a system to imitate means creating a function $f : S \rightarrow A$, which tries to predict the next action from the current states

Engineering good **task** imitators

This huge engineering and math challenge was the focus of last semester's course.

1. Large amounts of digital data and the ability to process it (fast computers, large storage, the internet, distributed computing, general purpose graphics processing units)
2. A “container” to store the imitation (most commonly the transformer), which builds on a century of research on artificial neural networks (which themselves copy several aspects of human brains)
3. A “recipe” for injecting the data into the container: minibatch stochastic gradient descent back by reverse-mode automatic differentiation

In this class, I assume you are familiar with these methods or will quickly learn about them

Useful **task** imitators

Applying these methods with different data sources yields lots of useful **task** imitators

AlphaFold 3: train on protein structure experiments to predict protein structure

- State: (protein description, protein structure so far), action: improved protein structure

LLM base models: train on internet scale natural language data

- State: tokens so far in the document, action: next token

Diffusion image generation: train on internet scale (image, text caption) pairs

- State: (text caption, image so far), action: improved image

CLIP: train on internet scale (image, text caption) pairs

- State: (image, text caption), action: embedding of both image and text to same space

Why might imitation be limiting? 1/3

Problem #1: the human limit

Unless the **task** behavior data we're imitating is optimal, our imitator will not be optimal

Three cases:

1. Data is optimal (e.g., protein structures from experimental results)
2. Data is mixed but contains optimal behavior
3. Data does not generally contain optimal behavior (e.g., internet)

Except in case #1: our imitator will be tied to the human limit 🙄 (but sometimes a little bit better than the best human due to consistency)

- We also have an engineering challenge to imitate the best behaviors and avoid influence of suboptimal behavior

Why might imitation be limiting? 2/3

Problem #2: **data scarcity**

We need *high quality* data to imitate

For different **tasks**, we have a less quantity and/or quality of data

- E.g., some languages we have almost no data
- E.g., physical world has been problematic (not that much digital data with joint angles, forces, etc., and hard to simulate well)

Why might imitation be limiting? 3/3

Problem #3: **differences between human and machine capabilities**

The imitator is imitating the output of another system (often a human)

Humans and machines have different hardware and software → the optimal machine approach to a **task** is not necessarily the same as the human way

- E.g., easy for a machine to master chess with massive computation, but this is impossible for a human. Imitating an expert human is a really bad way for a machine to play chess!

Imitation leads the machine to believe it has the same strengths and weaknesses as the human

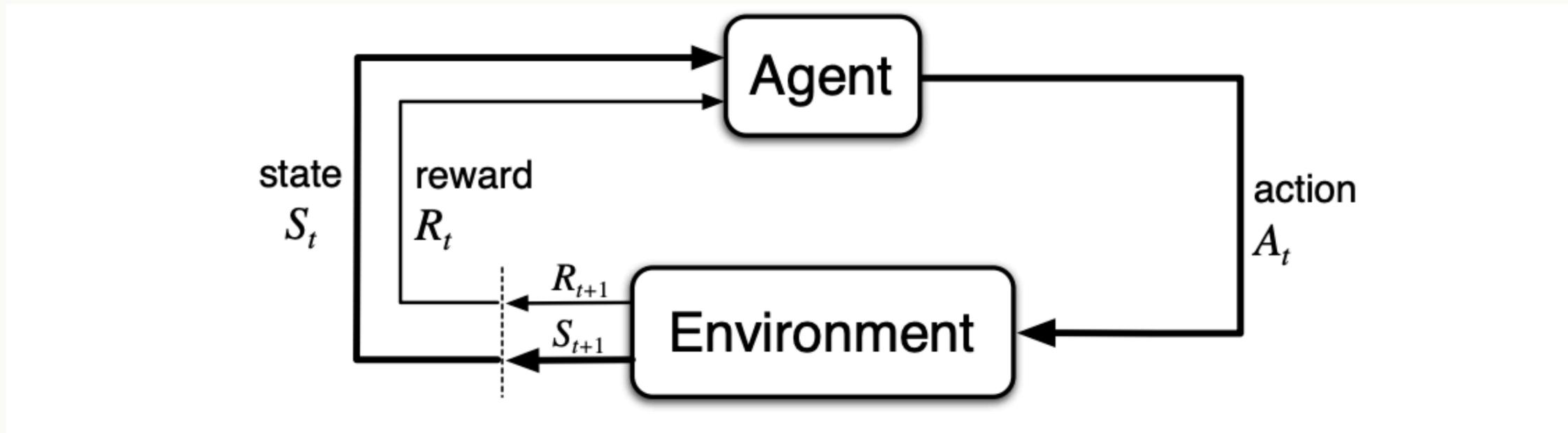
- Can easily make a completely unhuman mistake and end up OOD

Learning from experience

- Instead of imitating another system (like a human), let's consider this setup:
- Humans specify a **task** environment, which describes what actions are available to the AI, how the environment changes in response to the AI's actions, how much "reward" the AI gets (measure of success)
 - Humans specify container to hold what is learned, recipe for learning from experience

Soon, we will formalize these **task** environments as **Markov Decision Processes (MDPs)**

Reinforcement learning (RL)



Obligatory diagram of a RL agent interacting with the environment

- Any optimization in an environment* = RL according to Sutton and Barto
- Defined by goal, not method—don't need any actual reinforcement or even to interact with the environment 🤔
 - Note: S&B annexed all of optimal control and operations research as RL 🤔
 - I don't love this, but I think it's too conceptually messy to do otherwise so we're going with it (sorry, Yann LeCun, model predictive control is RL)

What's good about RL?

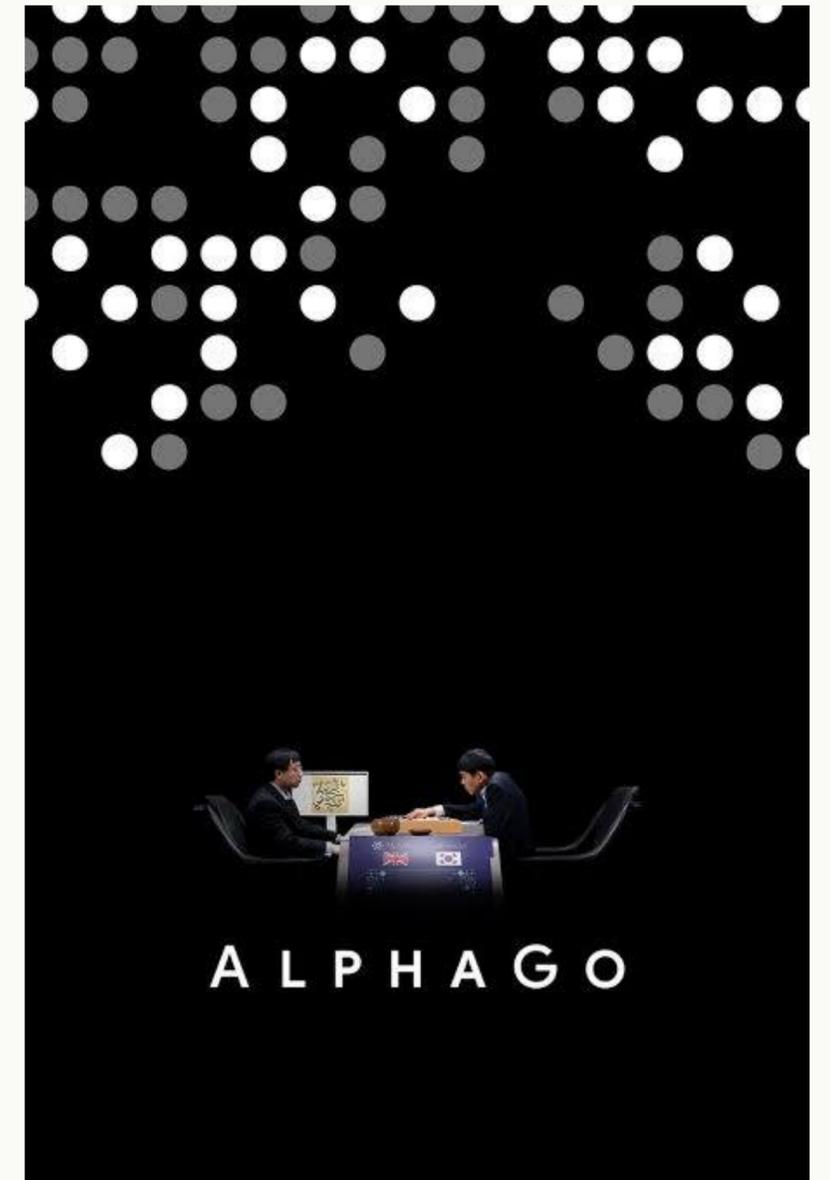
Recall three issues of imitation: (1) human limit, (2) data scarcity, (3) difference in capabilities

(1) and (3) are avoided entirely by RL systems

- By specifying what the desired **task outcome** is and not the desired **task behavior**, we allow the AI to surpass human capabilities and to do so however is most convenient

“Move 37” — Game 2 of AlphaGo vs. Lee Sedol

- Seemed like a mistake to human players, but was strategically brilliant
- This is what we want!



What's good about RL?

Recall, data scarcity = imitation needs high *quality data* to imitate

With RL, we don't need *high quality* data to imitate because the system learns without a teacher

However, we do need enough data to build an environment model to interact with, so data scarcity remains a huge problem in RL

- Data scarcity (especially across modalities) is one of the major blockers to AI today—there is good evidence that cross modality performance is poor for current systems

What's bad about RL?

It is **really hard** to learn without a teacher

For some **tasks**, we can might be able to “try everything”, e.g., games like chess/Go

- Then we have some hope of finding the best actions through exhaustive search

For many **tasks**, it is completely impossible to try everything

- There are just way way too many: e.g., physical movement, math proofs, potential sequences of words

Example: stuck in a box

You wake up inside a box

It is full of holes but they are all too small to escape through

There is alien experimenter outside, holding a clipboard

What do you do?



Stuck in a box

You likely want to get out of the box

You might as well assume there is a way out (otherwise, what can you do?)

You might guess that there are two possibilities:

1. There is some kind physical mechanism, like a button or switch that opens the box
2. You need to do, say, or think something to trigger the release mechanism that is unrelated to the physical box, like count to one thousand out loud'

Is there any hope to escape?

You are completely at the mercy of the experimenter

Your chance of escaping is entirely dependent on how hard the experimenter made the problem

Easy (minutes–hours): there is a largish physical button or switch you can feel around for

Medium (hours–days): there is a one-word trigger in a language you know

Hard (?): there is a very specific physical motion required or multiword phrase

Learning without a teacher is not, in general, possible

Humans have been able to learn a lot about the world, but

- It has taken a lot of people a very long time
- There are tons of things we still don't know

On the other hand, humans never had a teacher

- We learned all this stuff through experimentation
- Maybe an artificial system with a very different approach could learn stuff that would be very hard for us to learn and then explain it to us!

What can we do to make RL easier?

Three big ideas

1. Pick tasks where we can reasonably try everything (Go, Chess, Atari, etc.)
- The 2010s RL successes took this approach

2. Use mathematics to reduce the search space (convex functions, fixed point theory, etc)

- Most robotics today does not use experimentation at all, instead use mathematics (control theory) and limit to problems where control theory works

3. Use an imitative prior

Imitative priors

Compromise of the moment: use imitation to learn a distribution over “good” **task** behaviors, then use RL to extract the “best” behaviors from that distribution

By doing this, we dramatically reduce the search space, but also limit what we can learn (the behavior we learn must already be present in the base model)

Example: are bugs real? GPT3

Are bugs real? I mean, we've never seen any. But do butterflies turn into chrysalises like caterpillars do? I bet it's gross to have grubs crawling around inside you. Too bad bunny loaf doesn't like to play fetch. I'd be able to take him hiking

Variants

Node

I mean, we've never seen any. But do butterflies turn into chrysalises like caterpillars do? I bet it's gross to have grubs crawling around inside you.
Too bad bunny loaf doesn't like to play fetch. I'd be able to take him hiking

Do you find their legs gross? Visiting the bug house at London Zoo, the Thorne siblings will soon discover the importance of looking at the spectacular, wonderful and terrifying world that surrounds us. Cute and creepy ...does it get much better than that? I am now on a quest to find

If you were in a deserted island and were given one way to get food, would you ever eat a bug for survival? Bugs are the last thing you think to eat (i.e. if the events of Into the Wild were to actually happen), but oddly, it's one of the most

\n\nSome bugs seem to have been invented by artists of furniture and rugs. If you think that, you are absolutely correct. But, bugs such as ticks and fleas are also to be considered true, but don't take this fact too lightly as they can carry harm. Bugs can transfer

Do dinosaurs exist? Are there aliens running around? Absolutely. Some people see cryptic or strange pictures when gazing at the Sun or at the Moon. Do they really see strange pictures? No. It is simply that when they look at these things they tend to concentrate on things they want and want them to

model text-davinci-002 n 5 temp 1 tokens 60 Generate

Example: are bugs real? After RL

Are bugs real?

There is no one answer to this question as it depends on what you mean by "bug." If you are referring to insects, then yes, they are real. If you are referring to computer errors, then again, the answer is yes - they exist.

' are'	-0.001 / 99.91%
' mean'	-7.346 / 0.06%
' re'	-8.387 / 0.02%
' define'	-11.600 / 0.00%
' simply'	-12.239 / 0.00%
' consider'	-12.542 / 0.00%
' use'	-12.859 / 0.00%
' re'	-12.980 / 0.00%
' are'	-13.193 / 0.00%
' mean'	-13.527 / 0.00%
' Are'	-14.061 / 0.00%
' definition'	-14.336 / 0.00%
' defined'	-14.582 / 0.00%
' ARE'	-14.691 / 0.00%
' refer'	-14.791 / 0.00%

model text-davinci-002 n 5 temp 1 tokens 60 Generate

Variants Node

- \n\nThere is no universal answer to this question as it depends on what one considers to be a "bug". Generally speaking, most people would likely say that bugs are real, given that they can be observed and interacted with in the physical world.<|endoftext|>Although the use of the
- \n\nBugs are real, but not all bugs are insects. Insects are a type of bug.<|endoftext|>Mega
- \n\nThere is no definitive answer to this question as it depends on how you define "bug". In general, however, most people would say that bugs are real because they can be seen, touched, and heard.<|endoftext|>First I want
- \n\nThere is no one answer to this question as it depends on what you mean by "bug." If you are referring to insects, then yes, they are real. If you are referring to computer errors, then again, the answer is yes - they are real in the sense that they exist
- \n\nWhat do you mean by "bugs"?<|endoftext|>Never had life been more
- \n\nThere is no universal answer to this question as it depends on what is considered a "bug." Generally speaking, most people believe that bugs are real, although there is some disagreement over what actually constitutes a bug. Some people believe that bugs are tiny insects, while others believe that they are any

Imitative prior + RL

Imitative prior restricts generation to “plausible” completions

- For GPT3, completions are grammatical but still very random (from lots of different contexts)

RL causes the model to become extremely confident about what the next token is

- Paul Christiano: “For text-davinci-002 the goal is to have the model do what the user asked as well as it can, not to sample from possible worlds. For example, if the user asks “Is X true?” and the model's probability is 80%, the intended behavior is for the model to say “Probably” 100% of the time, not to say “Yes” 80% of the time and “No” 20% of the time.”
- I.e., maximize expected reward

The cost of the imitative prior

The RL procedure can only find the best actions that are present with some probability in the imitative prior (base model)

Intuitively, the best we can hope for is for the model to find the most useful parts of the human behavior distribution for each state

- Useful = successful in that environment
- Useful = compatible with the model's "brain"

The return of human limits

Cartoon: let's suppose we have a prior $\pi : \mathcal{S} \rightarrow \Delta A$, which is trained via imitation learning on a dataset of trajectories sampled from expert policies π_i

We then run a perfect* RL algorithm, maximizing expected reward, using π as the initialization, to produce π^*

π^* is not optimal because some actions may have zero probability in π

But π^* will be at least as good as all the individual experts and will generally be better, *by stitching together the best ideas of all the experts*

Stitching in practice

[728]	Aristotle, ChatGPT 5.2 Pro	6 Jan, 2026	● Full solution (Lean), using arguments similar to ● Pomerance (2015)
[729]	Aristotle, ChatGPT 5.2 Pro	8-10 Jan, 2026	● Full solution (Lean), using a modification of the solution to #728

Jan 6-10, 2026: first AI-generated solutions to Erdős problems
- Exactly what we would expect to see from “stitching”, transferring known arguments from one problem to another

After many many rediscoveries of human results that were not indexed properly

<https://github.com/teorth/erdosproblems/wiki/AI-contributions-to-Erd%C5%91s-problems>

Course structure

Three parts:

- ~5 lectures on RL at the start
- Student-led paper presentations
- Course project presentations

Part 1 of this class: lectures

Markov Decision Process (MDPs) theory: what are MDPs like from an optimization perspective, how can we solve them and with what kind of guarantees do we have?

Deep RL (2013–): how did deep learning transform RL? How do things change when use neural network architectures? *Less emphasis*

Policy gradient theory: the main point of this class—a powerful way to continue training a supervised deep network with RL

I will try to do this in 5 lectures (but could easily spend the semester on it)

Part 2 of this class: presentations

You will present a recent (2025-26) research paper to the class

- Presentations will be done in groups of 1–3
- Length: ~20 mins (OK if a little shorter or longer)

Each paper will have a discussant assigned. This person should also read the paper and prepare questions, comments and/or criticisms

- They will speak first after the presentation

A list of potential papers is on the class website, but I encourage you to **present a paper that is relevant to your course project**

More detailed guidance of what is expected in presentations is on the website

Part 3 of this class: projects

You will complete a course project (groups of 1-3). This is expected to have the form of a research paper, but need not be publishable (though quite a few were published last seminar)

Should relate to RL in some way (but can be a loose connection)

Ideally should relate to PhD research you're doing or want to do

Checkpoint: project proposal, due ~Mar 1, with what you plan to do (more details on the website)

Final presentations in ~April with final report due end of April

Final report grading is driven by completeness and correctness (e.g., did your approach make sense for your research question, did your evidence support your claims)

Presentations

Overall guidance: in my experience, the hardest thing is to hold the audience's attention

What I generally see is a large fraction of the audience tuning out in the first ~minute

Number #1 rule: do your best to make your presentation engaging

- Your impact = presentation utility \times number of people paying attention
- This always starts from answering the question “why should I care?” or “why is this worth my time?”

More presentation guidelines and advice on the course website

Attendance and participation

This is a small seminar class: attendance and participation are expected

The value we get out of the class is proportional to the sum of total contributions

We ask that you make positive contributions and avoid negative ones

- Positive contributions: questions, making an engaging presentation, sharing useful info with the class, being active on Teams

- Negative contributions: being on your phone/laptop (especially with people behind you), not paying attention

Evaluation

1% initial assignment

4% active participation in the class

20% paper presentation and discussion

75% for the final project:

- 20% project proposal
- 35% project write-up
- 20% project presentation

Initial assignment

We're trying Teams for discussion this class

Initial assignment will involve joining the Teams channel, messaging me on Teams to introduce yourself, tell me about your interests and what paper you might like to present

Due in two weeks (no class next week due to AAAI travel)

Worth 1% of your grade

Full details on course Carmen page

Statement on AI tools

You are explicitly allowed and encouraged to use any AI tools in this course, for any purpose

Here is what you get for free at OSU

- Github Education comes with a free Copilot subscription that lets you use the best coding model (currently, Claude Opus 4.5) for free via Copilot in VSCode
- Free Gemini 3 Pro access via OSU in chat mode

Non-OSU free things

- \$300 of free Google Cloud credits (Gemini 3 access via API)
- Antigravity (Google's agentic coding tool) has generous free limits

Where are AI tools good?

Recently, agentic coding tools have gotten a lot better. My opinion: most people should be using them extensively

- But they can trick you, don't trust them blindly
- My ranking of current tools:

Claude Code > Antigravity = Windsurf = Cursor = Codex > VSCode Copilot

AI tools are also good for:

- Finding related work (probably worse than your concerted effort, but much faster)
- Providing feedback on your plan or paper
- Executing simple math proofs
- Explaining papers/ideas/math in lots of detail
- Explaining why people care about some idea, when it seems dumb to you
- Explaining why people think some idea is bad, when it seems smart to you

Where are AI tools bad?

Training data cutoff: many papers we read are not in the training data of most models (Gemini 3 January, Opus 4.5 March, GPT5.2 August)

Models struggle to read PDFs of papers well, especially figures and tables

Models hallucinate what is actually in a paper if you refer to it only by title

- Can mitigate this by providing a PDF, or even better, a markdown or LaTeX version of paper in context
- Don't blindly trust "related work"

Proofs and maths are still fairly messy, but excellent in some areas

Bad at reasoning about time, cost, volume: what is a reasonable amount of stuff to do in the context of a course project? how many thinking tokens should task X require?

Bad at reasoning about the capabilities of models (can model X do Y?)

Other resources

We have around something like \$1.5k of OSC credits

- Please mention in your intro assignment if you want to be added to OSC

Working on a few other things

Questions?