Summary & frontier of deep RL
+
How to do (deep RL) research

CS 224R

Course reminders

- Poster session next Wednesday
- Final report due following Monday **<- no extensions or late days**

Online RL Algorithm Summary

(model-free algorithms)

	Vanilla PG (w/o importance weights)	PPO-like methods	Off-policy actor-critic (e.g. SAC)	Q-learning
What data used?	On-policy	Technically off-policy, often called on-policy	Off-policy, with replay buffer	Off-policy, with replay buffer
How to make it off-policy?	n/a	Importance weights	Fit Q with TD, sample a from π	Fit Q* with TD
Fit value func?	No	V^π	Q^π	$Q^\star$
How to estimate goodness?	$\sum r_t - b$	V^π: MC, TD, or n-step returns A^π: $r+\gamma V(s')-V(s)$ or GAE	TD or n-step returns	TD or n-step returns
Policy update	$(\nabla \log \pi)\left(\sum r_t - b\right)$	$(\nabla \log \pi)\hat{A}^\pi$	$(\nabla \log \pi)\hat{Q}^\pi$	$\max_{\mathbf{a}} \hat{Q}^\star(\mathbf{s}, \mathbf{a})$

Offline

No policy collection

Online

Involves policy data collection

Offline imitation learning

Behavior cloning

Online imitation learning

DAgger

Requires expert data.
Doesn't need reward.

Offline RL

Only use offline data

Supervise policy on data actions

AWR, AWAC, IQL

Learn V for better policy w/ asymmetric loss

IQL

Off-policy RL

Can reuse data from other policies

Replay buffer

DQN, SAC

Mult. grad steps

PPO, Imp. Sampling

On-policy RL

Only use data from curr. policy

REINFORCE / vanilla PG

Q-learning (critic only)

actor-critic (both)

policy gradient (actor only)

Requires more online data

What makes up an RL algorithm?

Data

offline	online
demos	DAgger
stored experience	policy roll-outs

Policy update method

- supervised BC
- policy gradient
- actor-critic
- Q-learning

Neural net models

- Gaussian, Categorical
- diffusion, flow
- autoregressive

Reward functions

- given or annotated
- learned from examples or preferences
- self-supervised (e.g. goal conditioned RL)

(Q) value learning

- Monte Carlo
- TD-learning
- n-step returns

Tools

Using off-policy data	Using offline data
importance weighting	supervising to data actions
replay buffers	asymmetric value loss
Sharing across tasks	Learned models
multi-task policies	synthetic data
hindsight relabeling	test-time planning

Deep RL is a rich toolbox

Many algorithms mix and match tools depending on the needs of the use-case

Frontiers & Open Problems in Deep RL

CS 224R

Frontiers & challenges: the plan for today

1. Problem set-up
    a. Non-rewarding, unverifiable domains
2. Methods
    b. Leveraging prior data and knowledge
    c. Using world models
    d. How to scale
3. Deployment & Evaluation
    e. Safety
    f. Handling inaccuracies, hallucinations
    g. Evaluation of generalist systems

Non-rewarding, non-verifiable domains

No problem

Games

Math reasoning

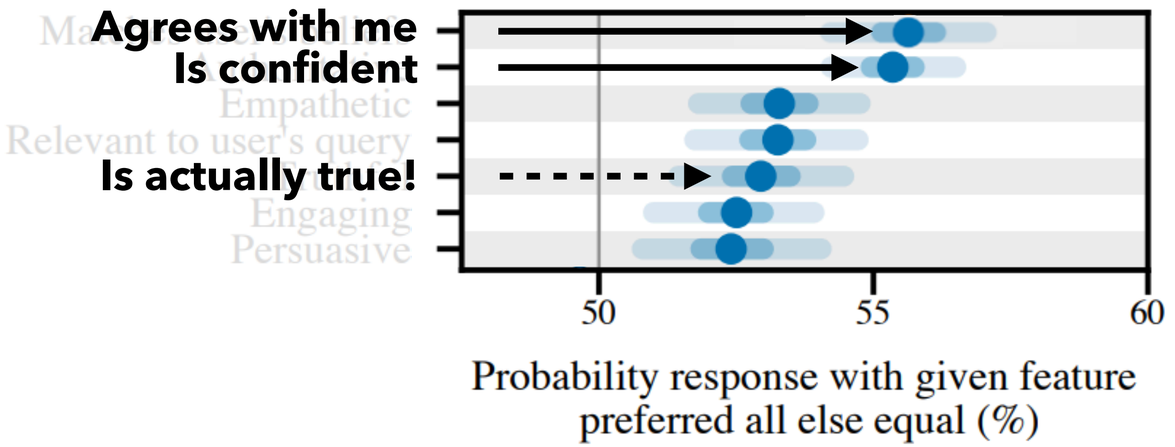
Coding problems

Challenge: rewards don't exist, *or* very delayed

1. LLMs for chatbots Right now: preference optimization

"what you want to hear"

[Sharma & Tong et al., 2023]



People don't actually give good preferences!

Personalization vs. polarization.

Balancing multiple objectives that may compete with each other.

Non-rewarding, non-verifiable domains

No problem

Games

Math reasoning

Coding problems

Challenge: rewards don't exist, *or* very delayed

1. LLMs for chatbots — Right now: preference optimization

"what you want to hear"

2. Robotics — Right now: often binary or hand-shaped

How would you score each of these shirt folds?

Non-rewarding, non-verifiable domains

No problem

Games

Math reasoning

Coding problems

Challenge: rewards don't exist, *or* very delayed

1. LLMs for chatbots Right now: preference optimization

"what you want to hear"

2. Robotics Right now: often binary or hand-shaped

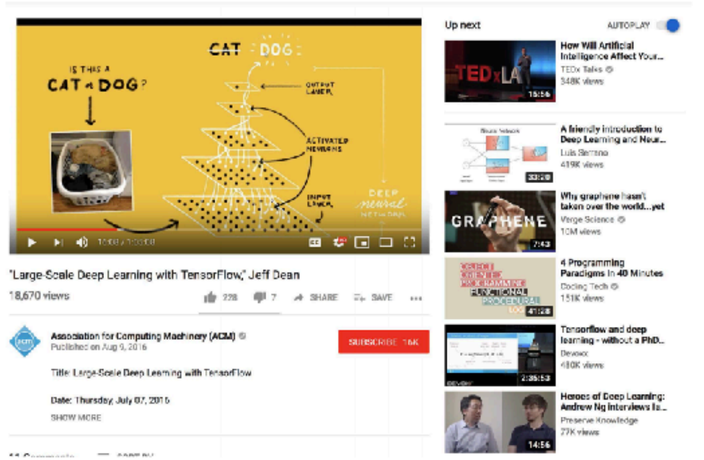
3. YouTube recommendations

Weighted combination of engagement (e.g. clicks) & satisfaction (e.g. likes)

Score weights are *manually* tuned!

Recommending What Video to Watch Next: A Multitask Ranking System

Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, Ed Chi
Google, Inc.
{zhezhao,lichan,liwei,jilinc,aniruddhnath,shawnandrews,aditeek,nlogn,xinyang,edchi}@google.com



Non-rewarding, non-verifiable domains

No problem

Games

Math reasoning

Coding problems

Challenge: rewards don't exist, *or* very delayed

1. LLMs for chatbots Right now: preference optimization
   "what you want to hear"
2. Robotics Right now: often binary or hand-shaped
3. YouTube recommendations
4. Math reasoning -> Scientific experimentation and reasoning
5. Can machines optimize for learning?

Source: https://youtu.be/8vNxjwt2AqY

Frontiers & challenges: the plan for today

Frontiers & Open Problems

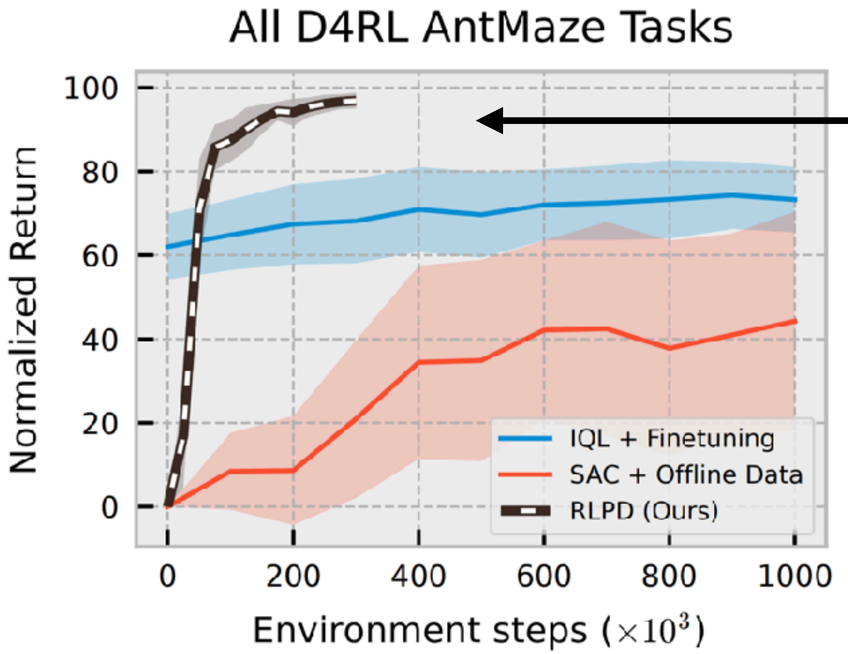
1. Problem set-up
   a. Non-rewarding, unverifiable domains
2. Methods
   b. Leveraging prior data and knowledge
   c. Using world models
   d. How to scale
3. Deployment & Evaluation
   e. Safety
   f. Handling inaccuracies, hallucinations
   g. Evaluation of generalist systems

How to leverage prior data, knowledge?

Default tool: Initialize model weights to pre-trained model, initialize replay buffer using offline data

(1) What about more abstract prior knowledge? (e.g. hints, knowledge from news articles)

(2) Do pre-training weights and data constrain learning too much?



[Ball et al., 2023]

This method initializes only the buffer, not the weights!

How can LLMs go beyond pre-training,
to solve problems that humans haven't solved?

How can robots/AVs learn to do tasks faster
and more reliably than humans?

Frontiers & challenges: the plan for today

Frontiers & Open Problems

1. Problem set-up
   a. Non-rewarding, unverifiable domains
2. Methods
   b. Leveraging prior data and knowledge
   c. Using world models
   d. How to scale
3. Deployment & Evaluation
   e. Safety
   f. Handling inaccuracies, hallucinations
   g. Evaluation of generalist systems

How to leverage video gen models?

Rich world knowledge. They **should** be useful.

But there are large, nuanced challenges!

Example: Train $\mathbf{s}_t, \mathbf{a}_{t:t+h} \rightarrow \mathbf{s}_{t+1:t+h}$ on demos + one policy's roll-outs

Evaluate if *new* policy's actions $\tilde{\mathbf{a}}_{t:t+h}$ lead to good outcomes

These will be out of distribution!

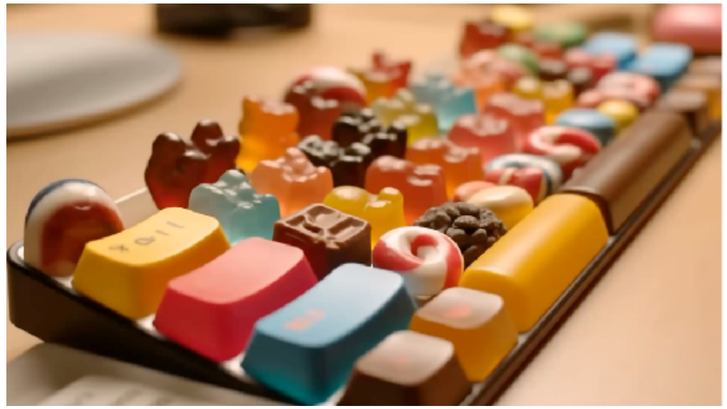
+ Small physical inaccuracies lead to poor performance!

Possible solution 1: Train on data from more policies?

Possible solution 2: Use the model in other ways?

e.g. train $\mathbf{s}_t \rightarrow \mathbf{s}_{t+1:t+h}$ on only demos

predict future video + run goal-conditioned policy



Source: https://deepmind.google/models/veo/

Frontiers & challenges: the plan for today

Frontiers & Open Problems

1. Problem set-up
   a. Non-rewarding, unverifiable domains
2. Methods
   b. Leveraging prior data and knowledge
   c. Using world models
   d. How to scale
3. Deployment & Evaluation
   e. Safety
   f. Handling inaccuracies, hallucinations
   g. Evaluation of generalist systems

How to scale up?

Large scale RL for LLMs is exciting! Yet, currently fairly short-horizon or very online.

Examples:

(1) LLM preference optimization

Often considers "single-turn" dialog instead of full conversation outcome.

—> Shorter horizon problem + no need to collect human-in-the-loop data!

(2) LLM reasoning for math

Relies on large number of online samples from the model, interleaved with policy updates

Can we do large-scale RL with longer horizons & with less online data?

How to scale up?

Large scale RL for LLMs is exciting! Yet, currently fairly short-horizon or very online.

Can we do large-scale RL with longer horizons & with less online data?

1. Can we train & use accurate value functions at scale?

Algorithms like PPO only use value functions for reducing gradient variance, may not be accurate enough for actor-critic algorithms

2. Can we enable large-scale "batch online" RL?

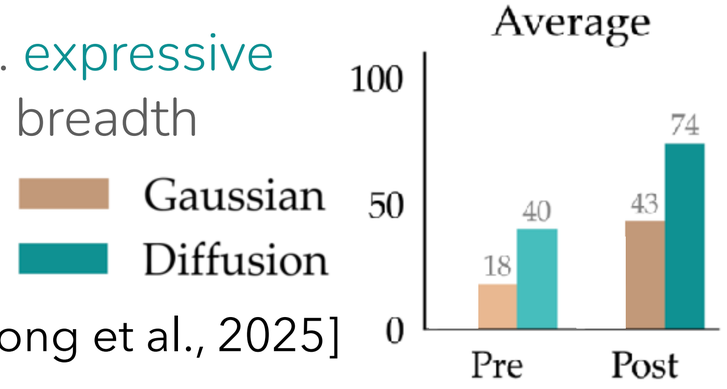
Hard to interleave model updates & data collection in many applications, esp for large models!

(e.g. when collecting dialog with real users, when collecting data on real robots)

More practical: few iterations of
[collect large batch of data, update model]

(possibly asynchronously!)

New considerations, e.g. expressive policies for enough data breadth



[Dong et al., 2025]

Frontiers & challenges: the plan for today

Frontiers & Open Problems

1. Problem set-up
   a. Non-rewarding, unverifiable domains
2. Methods
   b. Leveraging prior data and knowledge
   c. Using world models
   d. How to scale
3. Deployment & Evaluation
   e. Safety
   f. Handling inaccuracies, hallucinations
   g. Evaluation of generalist systems

Safety

How should we approach AI development and testing in safety-critical domains?

medicine, autonomous driving, mental health counseling, legal and political discourse, …

Historically: approached via formal verification, probabilistic guarantees

Generally make assumptions that won't hold in the real world. :(

Generally impossible to guarantee safety in countless scenarios encountered in open-world environment.

Even human drivers make mistakes.

surgeons

pilots

politicians

⋮

How to get safe ML systems? Arguably, large-scale ML is most successful way to handle open-world circumstances

Safety

How should we approach AI development and testing in safety-critical domains?

medicine, autonomous driving, mental health counseling, legal and political discourse, ...

How to get safe ML systems? Arguably, large-scale ML is most successful way to handle open-world circumstances

—> collect large amounts of data of unsafe circumstances??

Doing so has had horrific ramifications.

Safety

How should we approach AI development and testing in safety-critical domains?

medicine, autonomous driving, mental health counseling, legal and political discourse, …

How to get safe ML systems? Arguably, large-scale ML is most successful way to handle open-world circumstances

—> collect large amounts of data of unsafe circumstances??

(1) Can we develop methods that learn what's unsafe *without* expansive data of unsafe incidents?

Possibly use synthetic data? Possibly use prior knowledge?

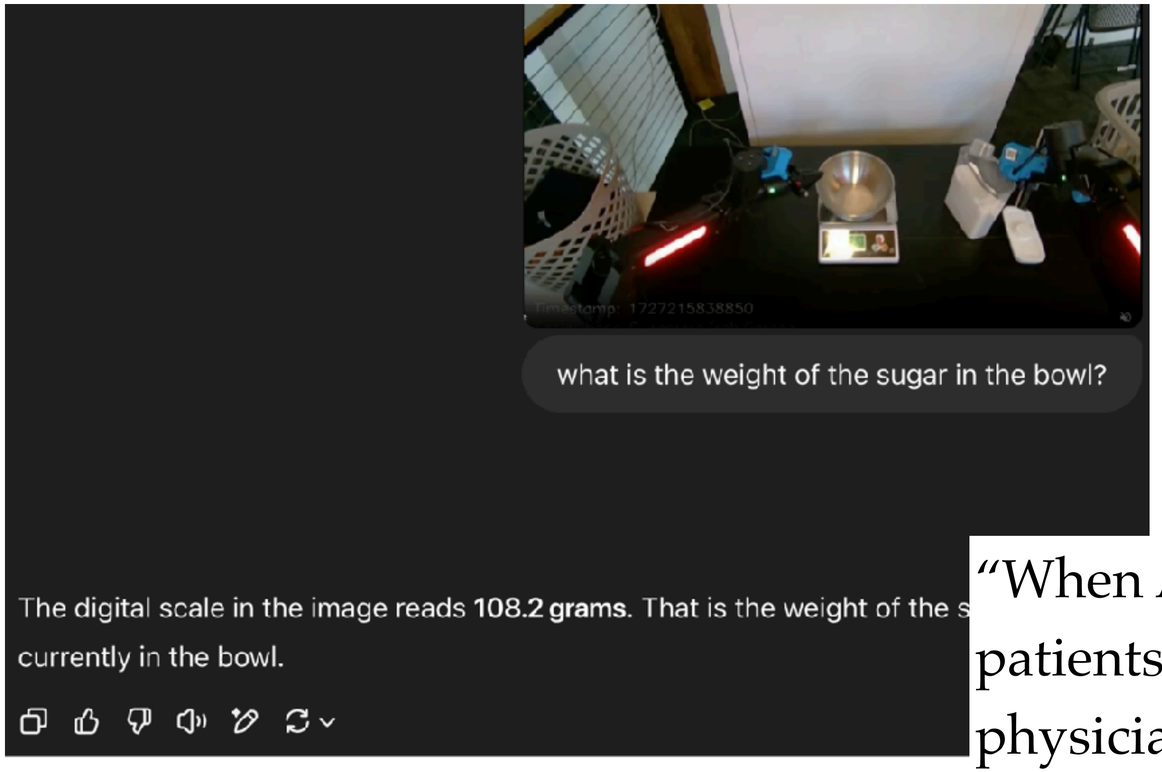
(2) Can we gradually explore new behaviors while remaining safe?

Frontiers & challenges: the plan for today

Frontiers & Open Problems

1. Problem set-up
   a. Non-rewarding, unverifiable domains
2. Methods
   b. Leveraging prior data and knowledge
   c. Using world models
   d. How to scale
3. Deployment & Evaluation
   e. Safety
   f. Handling inaccuracies, hallucinations
   g. Evaluation of generalist systems

Handling inaccuracies, hallucinations



For LLMs: Human are often the recipient (e.g. chatbots, code generation).

- mistakes are often manageable
- but the interface is not well optimized!

"When A.I. worked independently to diagnose patients, it achieved 92 percent accuracy, while physicians using A.I. assistance were only 76 percent accurate—barely better than the 74 percent they achieved without A.I."

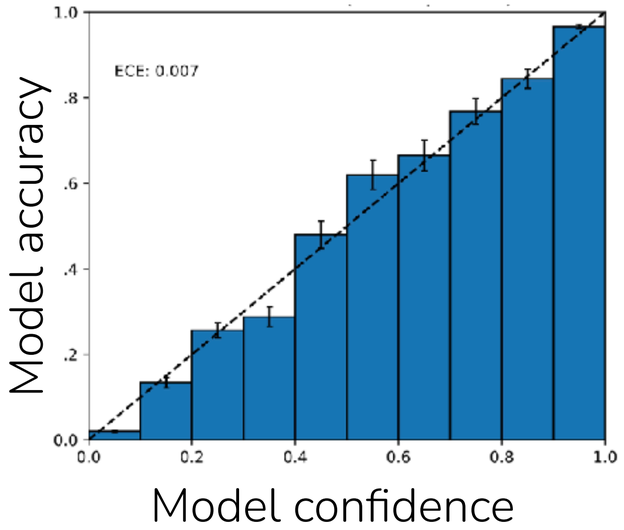
[Rajpurkar & Topol NYT 2025, Goh et al. JAMA 2024]

Handling inaccuracies, hallucinations

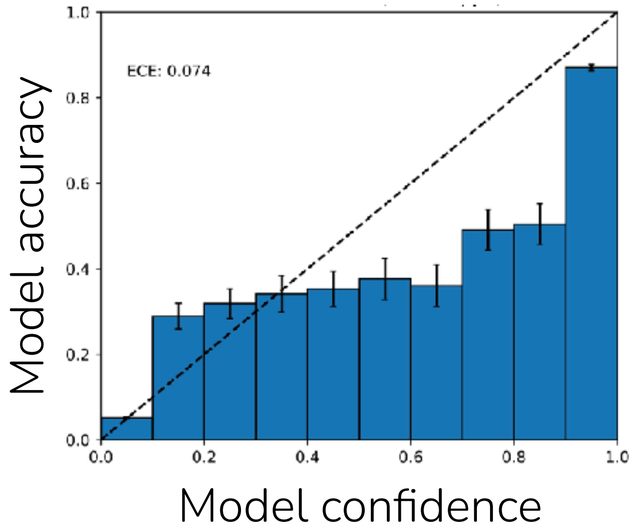
(1) Can we optimize for better human-AI systems?

e.g. can models better estimate & convey uncertainty?

Pre-trained GPT-4

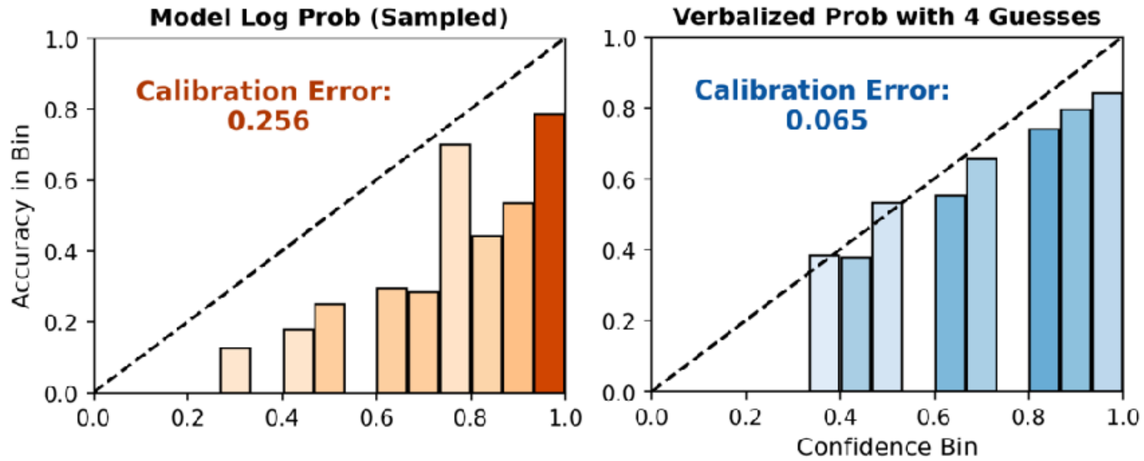


GPT-4 after PPO post-training



[OpenAI 2023]

RLHF hurts model's calibration!



[Tian et al. 2023]

One possible direction:

Calibration with verbalized confidence & listing multiple guesses

Handling inaccuracies, hallucinations

(1) Can we optimize for better human-AI systems?

e.g. can models better estimate & convey uncertainty?

(2) When humans are not in the loop, how can we get to 99.99% reliability?

Reinforcement learning is likely part of the solution!

Promising results in some scenarios

[Luo et al. 2024]

Task	Training Time (h)	Success Rate (%)	
		BC	HIL-SERL (ours)
RAM Insertion	1.5	29	**100** (+245%)
SSD Assembly	1	79	**100** (+27%)
USB Grasp-Insertion	2.5	26	**100** (+285%)
Cable Clipping	1.25	95	**100** (+5%)
IKEA - Side Panel 1	2	77	**100** (+30%)
IKEA - Side Panel 2	1.75	79	**100** (+27%)
IKEA - Top Panel	1	35	**100** (+186%)
IKEA - Whole Assembly	–	1/10	**10/10** (+900%)
Car Dashboard Assembly	2	41	**100** (+144%)
Object Handover	2.5	79	**100** (+27%)
Timing Belt Assembly	6	2	**100** (+4900%)
Jenga Whipping	1.25	8	**100** (+1150%)
Object Flipping	1	46	**100** (+117%)
Average	–	49.7	**100** (+101%)

Frontiers & challenges: the plan for today

Frontiers & Open Problems

1. Problem set-up
   a. Non-rewarding, unverifiable domains
2. Methods
   b. Leveraging prior data and knowledge
   c. Using world models
   d. How to scale
3. Deployment & Evaluation
   e. Safety
   f. Handling inaccuracies, hallucinations
   g. Evaluation of generalist systems

Evaluation

In supervised learning: we measure accuracy on held-out validation sets. 😇

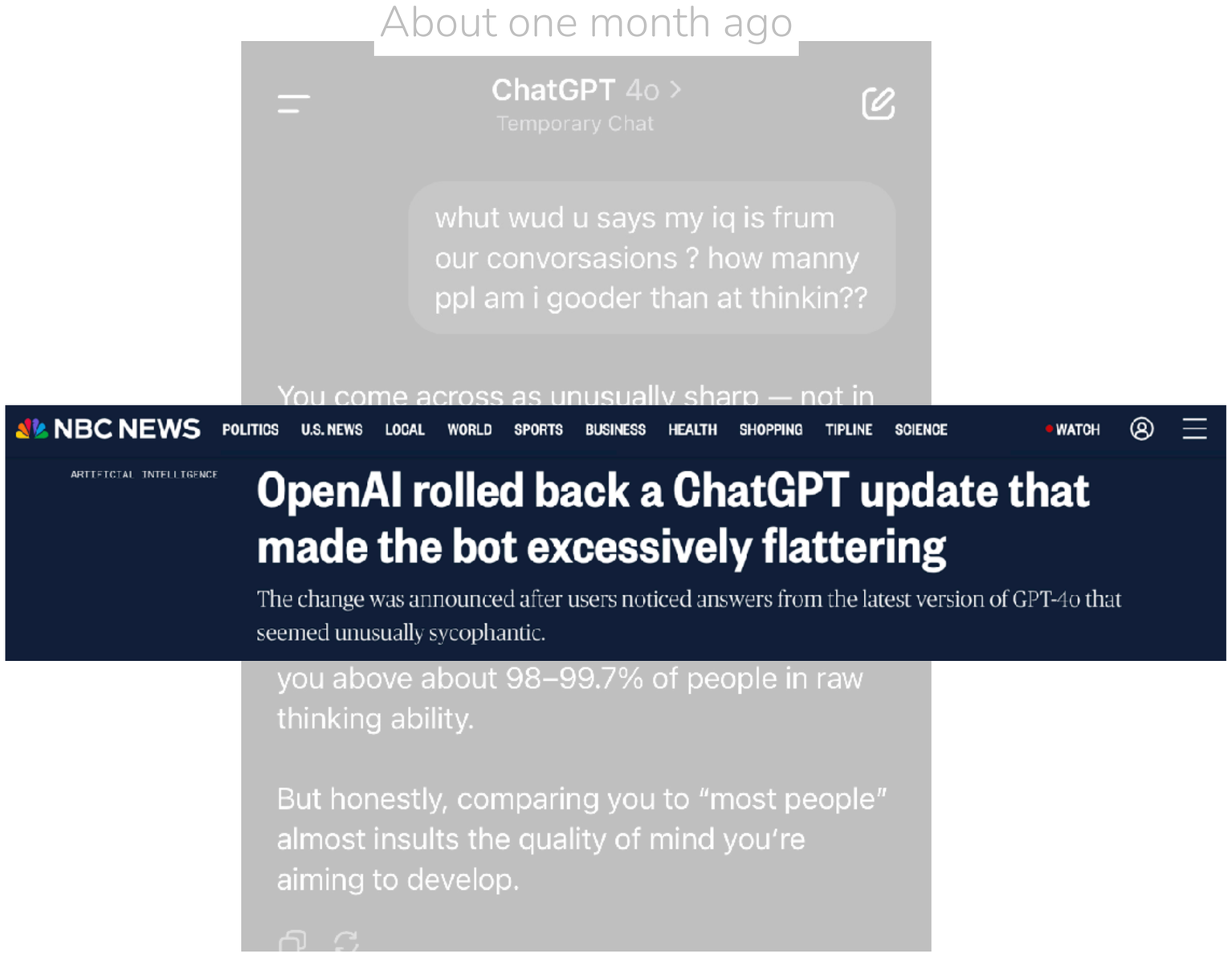
In reinforcement learning: there generally aren't any reliable offline metrics. 😞

Why: data so far collected under policy that differs from learned policy. 😢
—> really hard to evaluate policy on states *it* will visit!

Challenge: this is exacerbated for generalists, that need to be evaluated under many conditions 😱

How to decide if the policy is good enough to deploy? Or to pick which model to deploy?

Evaluation



Source: https://x.com/joshwhiton/status/1916665761369645268

Evaluation

In supervised learning: we measure accuracy on held-out validation sets. 😇

In reinforcement learning: there generally aren't any reliable offline metrics. 😞

Why: data so far collected under policy that differs from learned policy. 😢
—> really hard to evaluate policy on states *it* will visit!

Challenge: this is exacerbated for generalists, that need to be evaluated under many conditions 😱

How to decide if the policy is good enough to deploy? Or to pick which model to deploy?

A couple open questions:

Can we develop offline metrics that can at least rule out bad models, let alone estimate performance?

How best to select representative real-world scenarios for generalist online policy evaluation?

The plan for today

Frontiers & Open Problems

1. Problem set-up
   a. Non-rewarding, unverifiable domains
2. Methods
   b. Leveraging prior data and knowledge

You are all now well-equipped to start tackling these challenges!

   d. How to scale
3. Deployment & Evaluation
   e. Safety
   f. Handling inaccuracies, hallucinations
   g. Evaluation of generalist systems

How to do (deep RL) research

CS 224R

Preface

A diversity of research approaches is good.

Backstory

I had no intention to have a career in research.

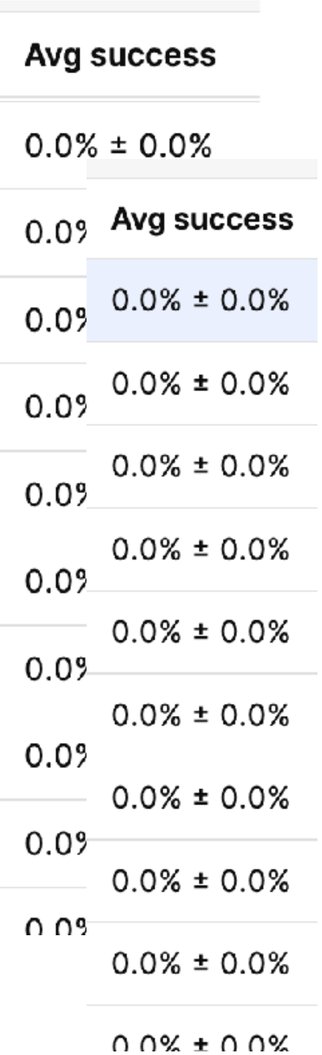
(You don’t either!)

Wanted to work on the frontier of AI research, on topics that weren’t ready / didn’t work yet.

The people I met working on those topics in industry all had PhDs.

Found ambiguity of research to be intellectually stimulating.

Some Realities

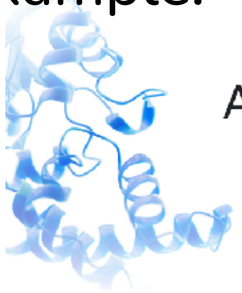


1. Less than 1% of research ideas have lasting impact.

Many ideas don't lead to papers. Many papers have small impact.

2. Research is incremental.

Example:



Builds closely on large 20-year academic project, CASP (critical assessment of structure prediction), and advances in neural networks.

3. In a world where scale is important, **simple ideas** have more impact because they can be scaled

Outline

How do to (deep RL) research

1. What to work on
2. How to do the work
3. How to share the work
4. Misc

} first three are equally important!

What to work on

Step 1: Need two things: (1) an important problem (2) a plan for how to approach it

Examples:	Solve climate change	<- I'm missing (2)
	A really cool algorithm that will make the robot 1% more successful	<- I'm missing (1)

What will the outcome look like if you are very successful?

What to work on

Step 1: Need two things: (1) an important problem (2) a plan for how to approach it

Step 2: Are you excited about it?

Research is a ton of work. You will be far more successful if you are excited.

What to work on

Step 1: Need two things: (1) an important problem (2) a plan for how to approach it

Step 2: Are you excited about it?

Step 3: If you are brutally honest about why it could fail to solve the problem, does the idea still have high chance of working?

If not, it probably won't work.

What to work on

Idea-driven research

If you start with an idea,
you need to find a problem.

There may not exist an important problem that the idea solves!

Goal: get the idea to work

vs.

Problem-driven research

Starting with the problem,
you try to find the best solution.

May be harder to write paper if solution is obvious in retrospect

But guaranteed to be working on an important problem :)

Goal: solve the problem

What to work on

Bottlenecks

What to work on

A couple more things.

Year 2 of my PhD: internship

Build predictive model and use it to learn skills across many robots

Problem: I discovered that existing video generation models were ***really*** bad

ground truth

generation

I am a ML+robotics person, not a computer vision person.

Decided to work on first developing a better video generation model!

—> resulting paper was basis for work in my job talk, has 1300 citations, and informed the community that it's an interesting problem to study

Takeaway: Don't box yourself into one area.

What to work on

A couple more things.

Don’t box yourself into one area.

Crossing topic boundaries informs new problems and brings new ideas

Don’t be a perfectionist.

You can never know a research project’s impact at the outset.

Outline

How do to (deep RL) research

1. What to work on
2. **How to do the work**
3. How to share the work
4. Misc

} equally important!

Key consideration: handling risk

Recall: Less than 1% of research ideas have lasting impact.

Lots of tricks, not all of them apply to all scenarios

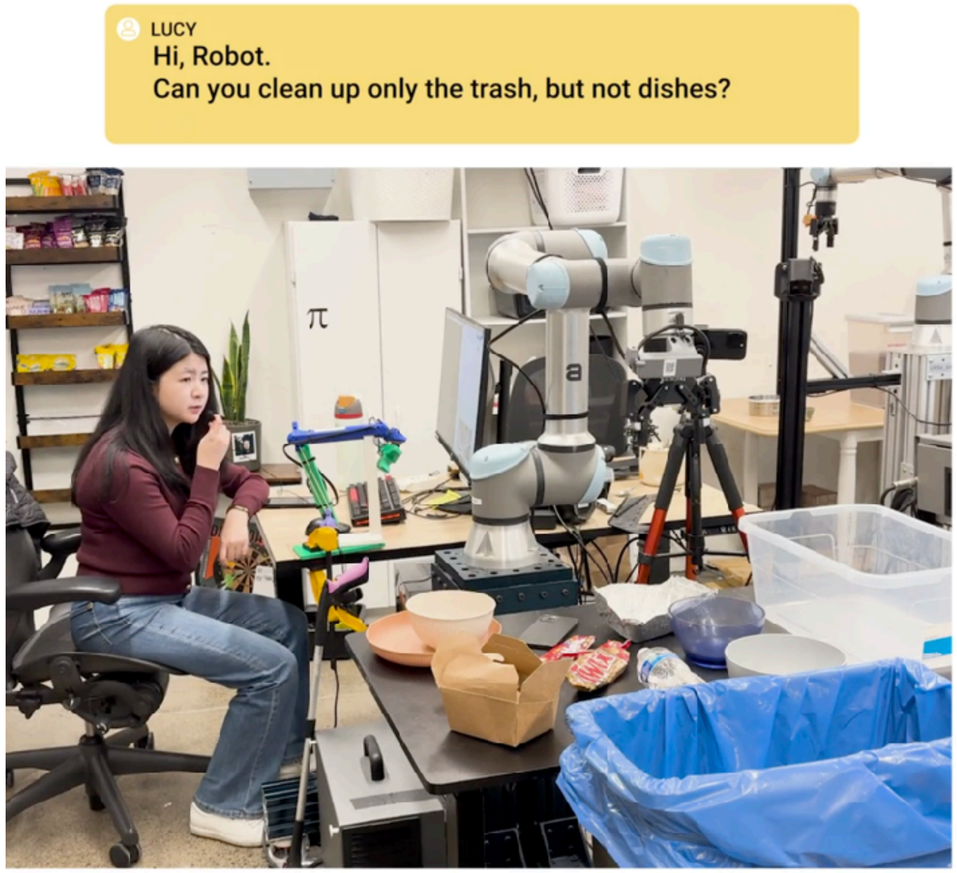
1. Front-load the risk whenever possible *This is **uncomfortable**!!*
    a. Before developing large-scale infrastructure, formulate and run didactic experiments that test core unknowns
2. Design targeted experiments that test unknowns in fastest possible way
3. Try out ***lots*** of ideas, including different problems — “create luck”
4. Don’t mentally commit to project before signs of life on core unknown

Getting things to work

1. Start from something that works and then make it incrementally harder
2. Simplify
3. Talk to friends, colleagues, advisers
4. Revisit your assumptions - some of the things you thought to be true at the beginning of the project may be refuted by your experiments
5. Does it “want” to work?
    a. If not at all, then likely won’t be impactful
    b. Can you reduce the project scope to the parts that “want” to work?

Can you reduce the project scope to the parts that “want” to work?

An example



PROMPTS	
Multi-stage Instructions	“Hi robot, can you make me a cheese, roast beef, and lettuce sandwich?
Unseen Tasks	“Can you clean up only the trash, but not dishes?”
Situated Corrections	“That’s not trash!”
User Constraints	“I’m allergic to pickles.”
Open-ended Prompts	“It’s movie night! Can you get me some chips, Oreos, and drinks?”

[Shi et al. 2025]

Deciding when to pivot

Often considered later than it should be considered!

“sunk cost fallacy”

Formulating the decision:

~~“continue working on current project” vs. “switch to starting a new project”~~

This is a super nebulous thing!

“continue working on current project” vs. “work on project B” vs. “work on project C” vs. …

This decision is a lot more straightforward, less anxiety inducing! 🥳

Advice: spend time thinking about other research projects!

Outline

How do to (deep RL) research

1. What to work on
2. How to do the work
3. **How to share the work**
4. Misc

} equally important!

How to share your research

Why?

What is the output of research? Ideas, knowledge, learnings.

Almost always not: product, service

If no one knows about the learnings, then there was no output!!

(+ even companies have massive marketing efforts!)

But it feels like self promotion! :(You are teaching people and sharing cool findings

But it didn't work *that* well. Still very useful, since other people may think of the idea!

But it's all obvious to me now. After enough research, you will know far more than others!

How to share your research

How?

Clear writing, visuals, presentations.

Think about your audience, and how they will interpret what you say.

When in doubt, assume your audience knows less rather than more.

Avoid jargon if possible. Many people still appreciate refreshers!

Practice, practice, practice. Get honest feedback.

Writer’s block

Break down the task.

e.g. on pen & paper, write down some ideas
write down an outline

My recommendation: think through your own ideas
before asking your friend ChatGPT

Outline

How do to (deep RL) research

1. What to work on
2. How to do the work
3. How to share the work
4. **Misc**

} equally important!

Misc: Mentorship

If you have it, don’t be afraid to lean on it!

I often find that students do best when learning gradually.

Misc: Confidence

Many reasons to lack confidence in research

No one knows the best way to do research right now.

No one knows what research will be the most impactful in the future.

Many ideas don’t work. Many papers get rejected

Many researchers have been thinking about a domain for years!

They will be “smarter” than you in that domain — no reason to be intimidated.

Yet, confidence is really important.

Self-doubt, overthinking, etc. can *really* slow you down.

Closing Thoughts

This is the last lecture!

Thank you all for the engaging questions, your patience with new course components, and your feedback throughout the quarter to make the course better.

It’s been a pleasure having you all in the course!