Last Time:

- [ ] game theory

This Time:

- [ ] alignment

Lecture 12

HRI, FALL '25

Andrea Bajcsy

What is AI Alignment?

You may have heard of this term "alignment" when discussing new "foundation models" like ChatGPT / Gemini or possible "robotics foundation models", but it's conception dates as far back as 1960 when AI pioneer Norbert Wiener described AI alignment as:

> "If we use, to achieve our purposes, a mechanical agency with whose operation we cannot interfere effectively .... we had better be quite sure that the purpose put into the machine is the purpose we really desire."

— Wiener, 1960. "Some Moral and Technical consequences of Automation."

$\boxed{\text{value alignment}}$ is the process of developing & deploying AI systems in a way that aligns with human values & goals.

! value alignment is fundamentally a multi-agent problem btwn. the AI / robot and the human – who determines what the objective is.

Famous examples:

1. Coast Runners Game (Amodei & Clark, 2016)
2. lego stacking Manipulator (popov et al, 2017)
3. Deceptive dexterous hand (Christiano et al, 2017)
4. Exploiting simulation bugs (Code Bullet, 2019)

How do we align an AI system?

There is no one algorithm or tool that can "solve" alignment, but one popular paradigm is called $\boxed{\text{reinforcement learning from human feedback}}$ (RLHF)

↳ key component of current LLMs (e.g. ChatGPT / Claude / Bard)

current Autonomous Driving predictors / planners ← see: Cao et al. "RLHF for Realistic Traffic Sim." ICRA 2024

next-generation generative robotics policies

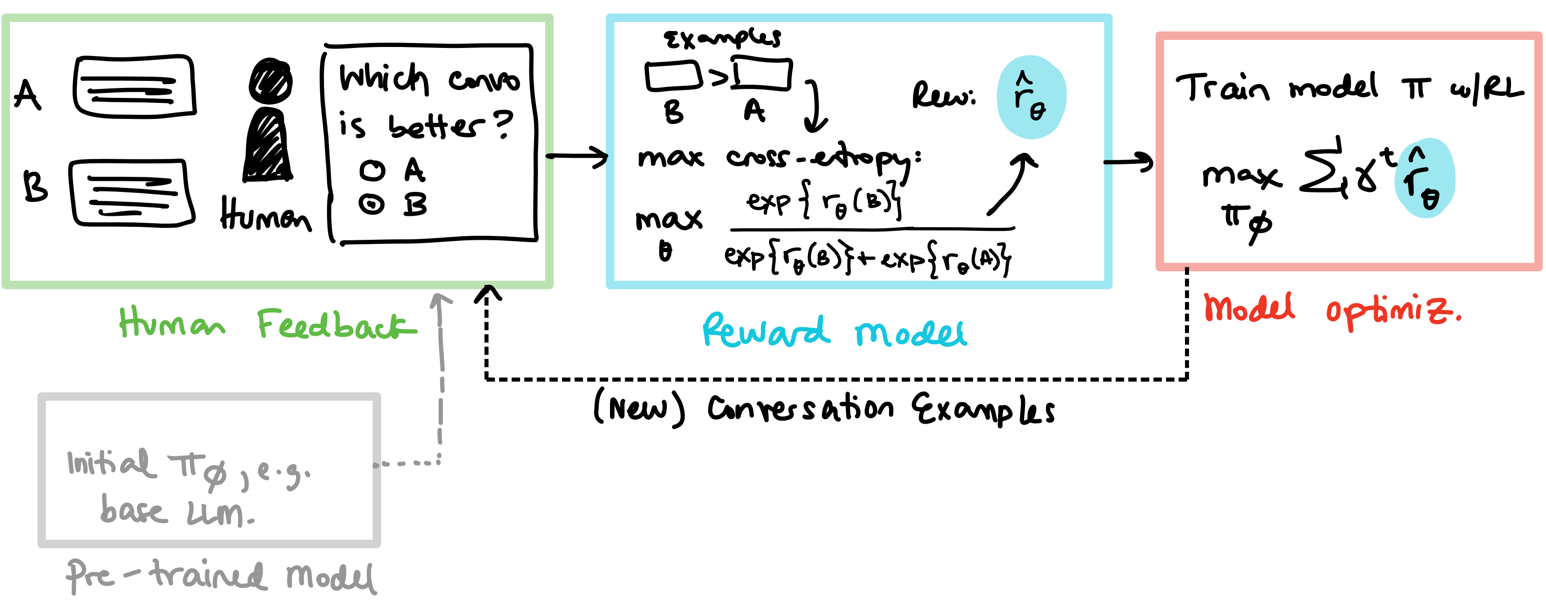
↳ see: Tian et al. "Maximizing Alignment with Minimal Feedback"

BUT, has roots in preference theory from economics w/ early applications in HCI and RL.

RLHF has three (repeated) steps:

1. feedback collection
2. reward modeling
3. model optimization

example: LLM chatbot RLHF w/ Binary Preference Feedback



Mathematically, the first step involves collecting examples from the "base model":

$$x_i \sim \pi_\phi$$

ex. a batch of 1+ generations from the model, like a complete convo., or a denoised action chunk in robotics FMs.

ex. LLM models: $P(\tau_i \mid \tau_{<i})$ next token pred. given prev. generations.

Let the human $\mathcal{H}$ have desires consistent with the unknown reward $r_{\mathcal{H}}$. Let the feedback we get from them be modelled by

$$y_i = f(\mathcal{H}, x_i, \epsilon_i)$$

human's feedback (e.g. their preference of B over A)

human

example generations (e.g. $x_i = (\text{convo A}, \text{convo B})$)

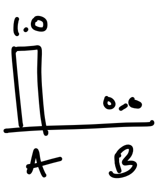
noise to feedback y_i

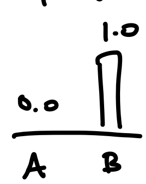
Next, we fit a reward model $\hat{r}_\theta$ with the feedback data: $\{(x_i, y_i)\}_{i=1}^N$ to approximate evaluations from $\mathcal{H}$ as closely as possible:

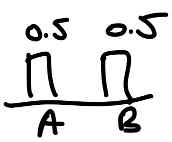
$$\hat{r}_{\theta^*} = \underset{\theta}{\text{minimize}}\ \mathcal{L}(\mathcal{D}, \theta) = \sum_{i=1}^{N} \underbrace{\ell(\overbrace{\hat{r}_\theta(x_i)}^{\hat{y}_i}, y_i)}_{\text{"suitable loss for this data"}} + \underbrace{\lambda_r(\theta)}_{\text{regularizer}}$$

ex. if $x_i = (\text{convo A}, \text{convo B})$

y_i = distribution over $\{A, B\}$ indicating which user preferred $= \mu$

$A \succ B \Rightarrow \mu :=$ 

; $B \succ A \Rightarrow \mu :=$ 

$A \approx B \Rightarrow \mu :=$ 

$A \neq B \Rightarrow y_i \notin \mathcal{D}$

"incomparable"

comparison discarded

We will model human judgements of preferring a convo as exponentially more likely the higher the reward:

$$P(A > B \mid \theta) = \frac{\exp\{\hat{r}_\theta(A)\}}{\exp\{\hat{r}_\theta(A)\} + \exp\{\hat{r}_\theta(B)\}} \qquad (*)$$

! Similar to the Boltzmann Rationality model, but now its over pairwise preferences rather than state traj. or actions, etc.

Formally $(*)$ is called the Bradley-Terry Model (B & T, 1952) and its a specialization of the Luce-Shephard choice rule (Luce 2005, Shephard 1957) to preferences over sequences/ trajectories.

We can now choose a loss function with this probabilistic model:

$$\ell(\hat{r}_\theta(x_i), y_i) := -\Big[\overbrace{\mu(A) \log P(A > B \mid \theta)}^{\text{if } A>B} + \overbrace{\mu(B) \log P(B > A \mid \theta)}^{\text{if } B>A}\Big]$$

↖ from our dist

Finally, we optimize the model with RL! Optimize π_ϕ model params:

$$\phi^*_{new} = \max_{\phi_{new}} \mathbb{E}_{x \sim \pi_{\phi_{new}}} \Big[\hat{r}_{\theta^*}(x) + \underbrace{\lambda_p(\phi, \phi_{new}, x)}\Big]$$

regularizer like divergence btwn. prior distribution π_ϕ & the new $\pi_{\phi_{new}}$

$$D_{KL}(\pi_{\phi_{new}} \,\|\, \pi_\phi)$$

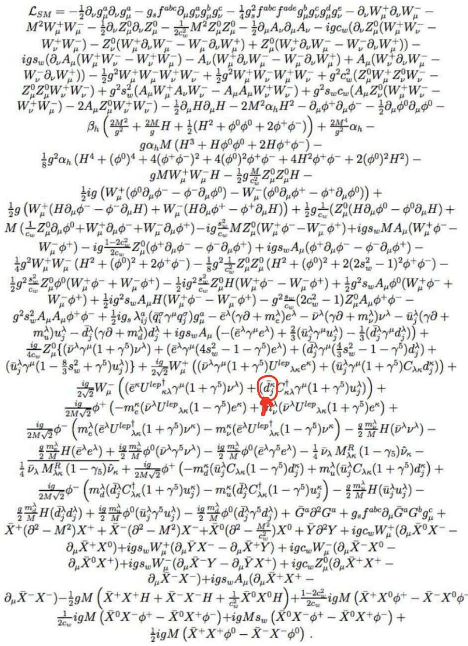
Challenges :

- Human Data

→ Humans struggle to evaluate difficult tasks well

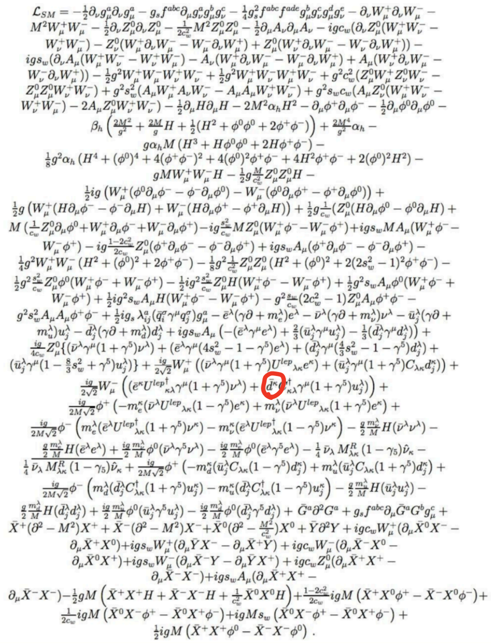
The "Standard model Lagrangian" of particle physics

Sample A



$\succ$

Sample B



Q which is more "correct"?

→ Human raters (across bgs, countries etc.) disagree

Is there a **SMILE** in this image?

YES but …

Canada		
YES	NO	DNK
40%	40%	20%

India		
YES	NO	DNK
70%	30%	0

USA		
YES	NO	DNK
50%	0	50%

adversarial example from the CATS4ML data challenge
https://github.com/google-research-datasets/cats4ml-dataset

from: Lora Aroyo, 2023. "The Many Faces of Responsible AI", Neurips Keynote

→ RLHF suffers from tradeoff btwn. the richness vs. efficiency of the feedback types.

↳ e.g. Comparisons : $y_i := A > B$

Scalar : $y_i = 0.35 \in \mathbb{R}$. ← more expressive! but poorly calib.

Label : $y_i \in \{Y_1, Y_2, \dots Y_M\}$ ← low effort but can suffer from choice set misspecification with incomplete labels.

Correction: $y_i = \Delta x_i$ ← higher effort

Language: $y_i = x$ ← utterance is easy, but imprecision of speech + cross-cultural diffs. make it hard

⋮ more??

- Reward learning

→ an individual's values are difficult to represent with a reward function

↳ e.g. human feedback can depend on contextual factors not easily accounted for in the training data (time-varying rewards, pedagogic behavior, etc.)

→ a single reward cannot represent a diverse society of humans (e.g. Smile example from above)

↳ can end up disadvantaging certain groups.

→ reward models can misgeneralize to poor reward proxies, even from correctly-labeled training data

↳ learned models are prone to causal confusion and poor out of distribution generalization

- Model optimization:

→ It is still hard to optimize models / do RL!

→ Policies can perform poorly @ deployment even if rewards seen @ training were perfectly correct.