Declarative characterizations of direct preference alignment algorithms

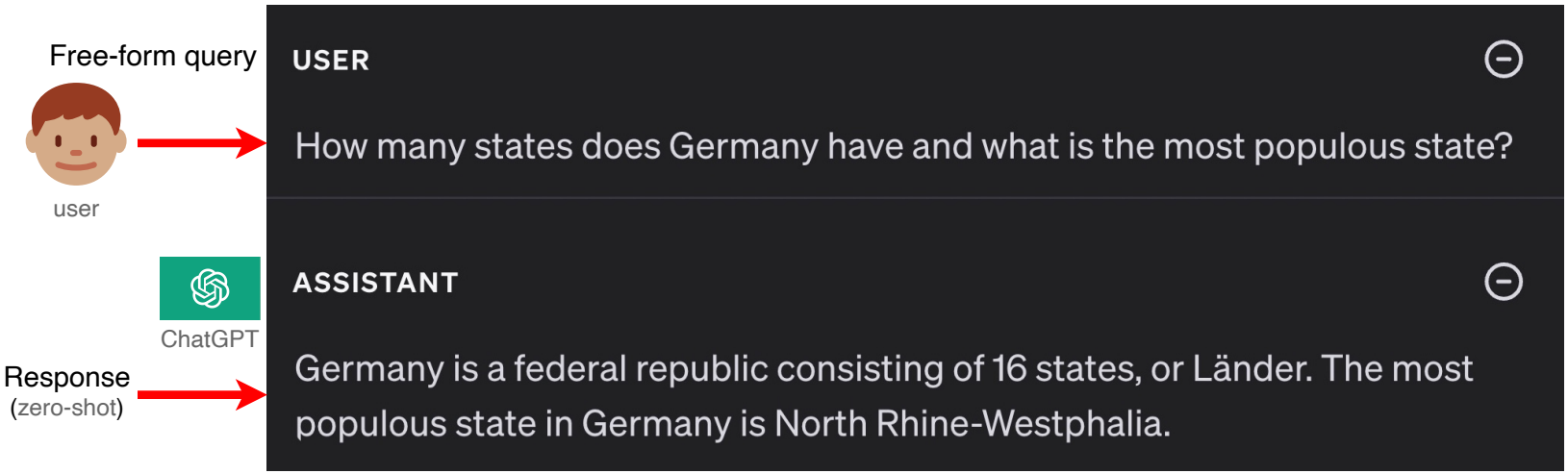
Kyle Richardson[α], Vivek Srikumar[β], Ashish Sabharwal[α]

Allen Institute for AI (AI2)[α]
University of Utah[β]

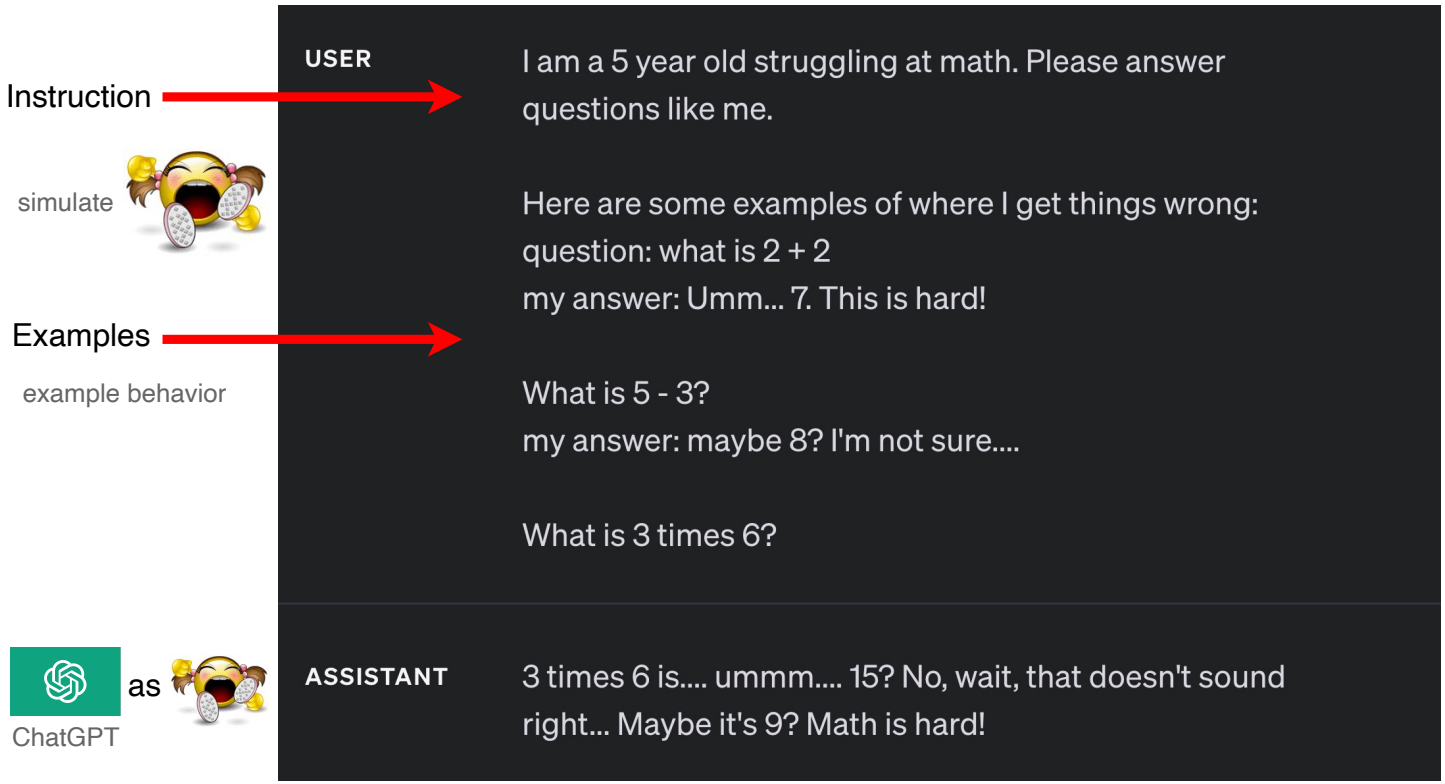
December 2024

Thoughts on language models, where we are

Large language models as general-purpose reasoners

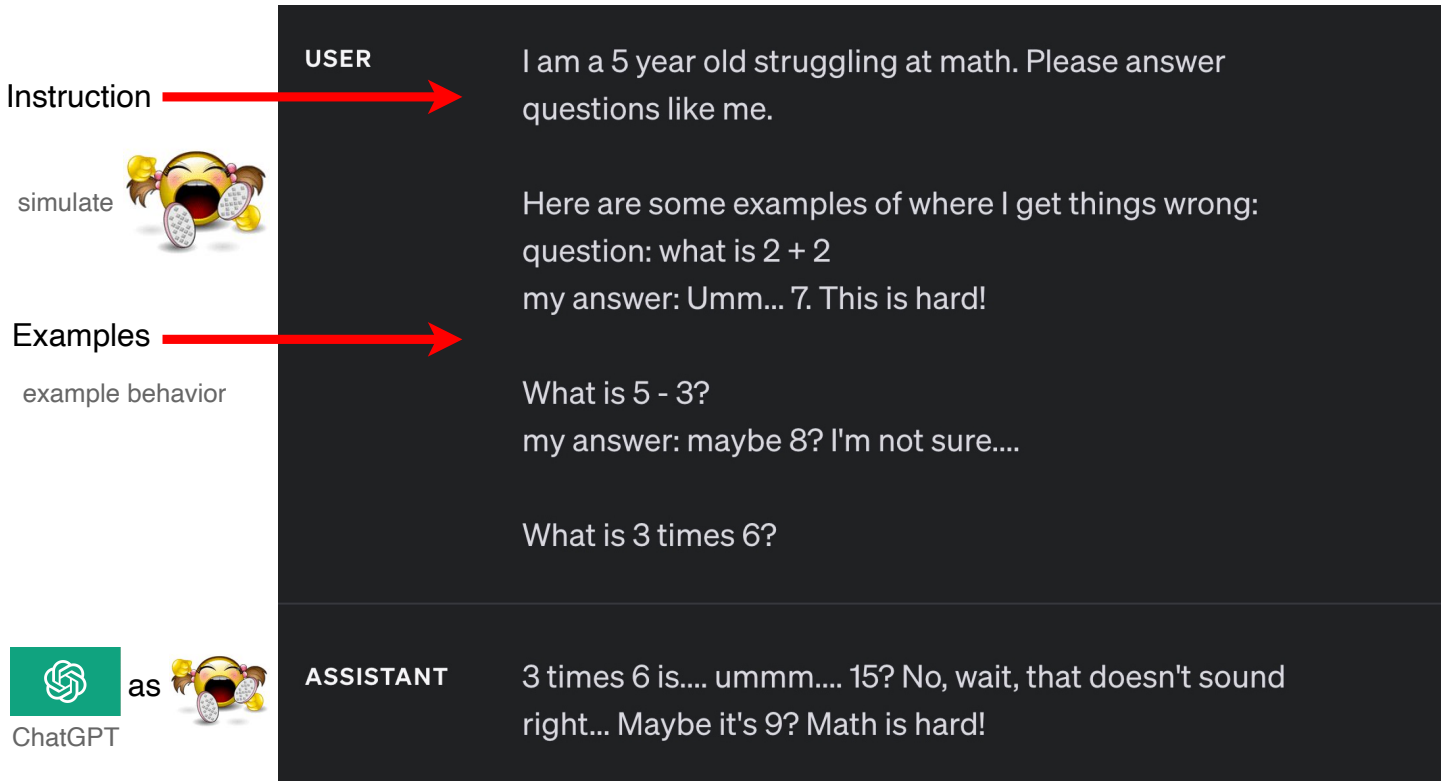


Getting information from models directly via prompting



Traditional learning coupled with **in-context leaning** (no parameter updates, just new examples in input).

Getting information from models directly via prompting



Has greatly lowered the bar to entry, made models more accessible, large part of recent success,; due to large-scale pretraining, RLHF and alignment.

Models have far exceeded expectations

(examples from me and my group)

Language models as agent simulators

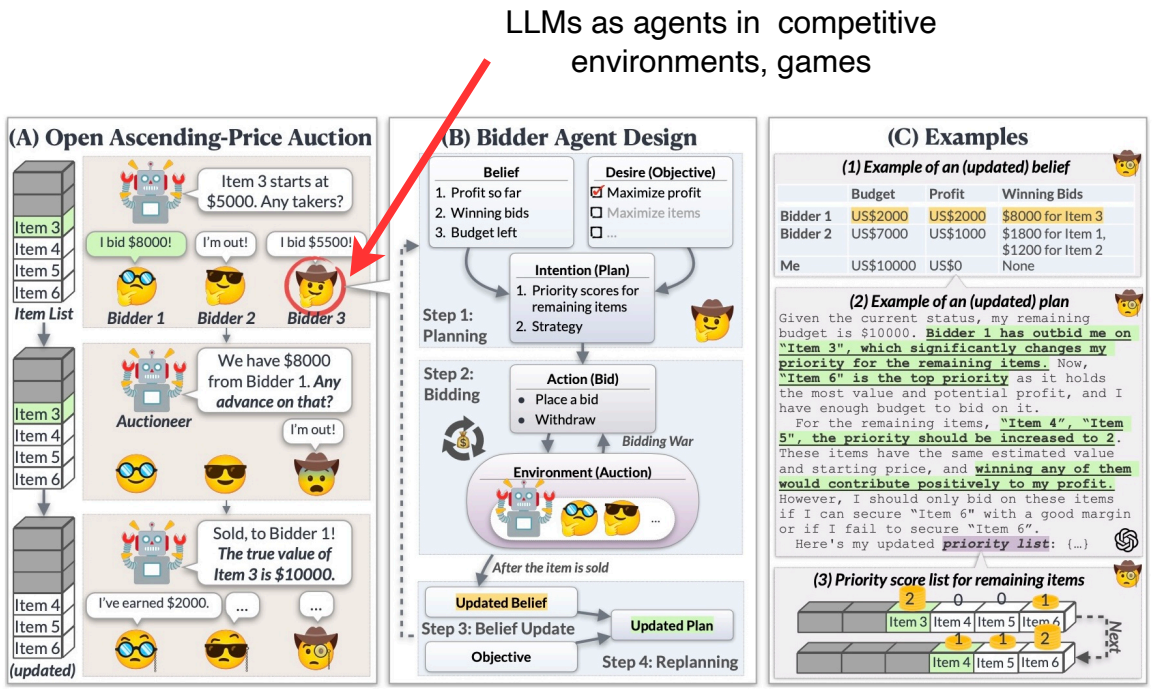


Figure 1: An illustration of AUCARENA: (A) shows a multi-round, ascending-price auction with an auctioneer announcing the highest bids, where bidders publicly decide after private reasoning. (B) presents a bidder agent structure using the Belief-Desire-Intention Model, involving planning, bidding, belief updating, and replanning, where beliefs and plans are adjusted with each auction development. (C) offers instances of updated beliefs and plans, illustrating a bidder allocating priority scores to items post-reasoning.

- Can we use LMs to simulate complex social dynamics? (Yang et al., 2024; Chen et al., 2023), (Zhang et al., 2024)[**ACL**]

Language models as agent simulators

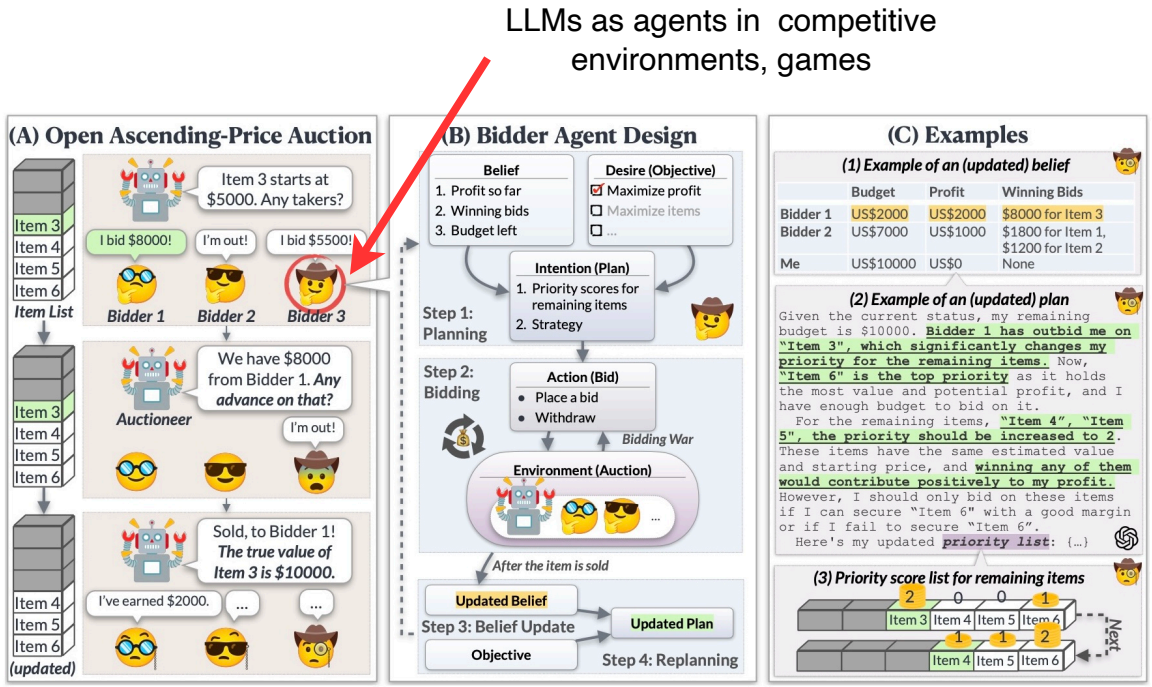
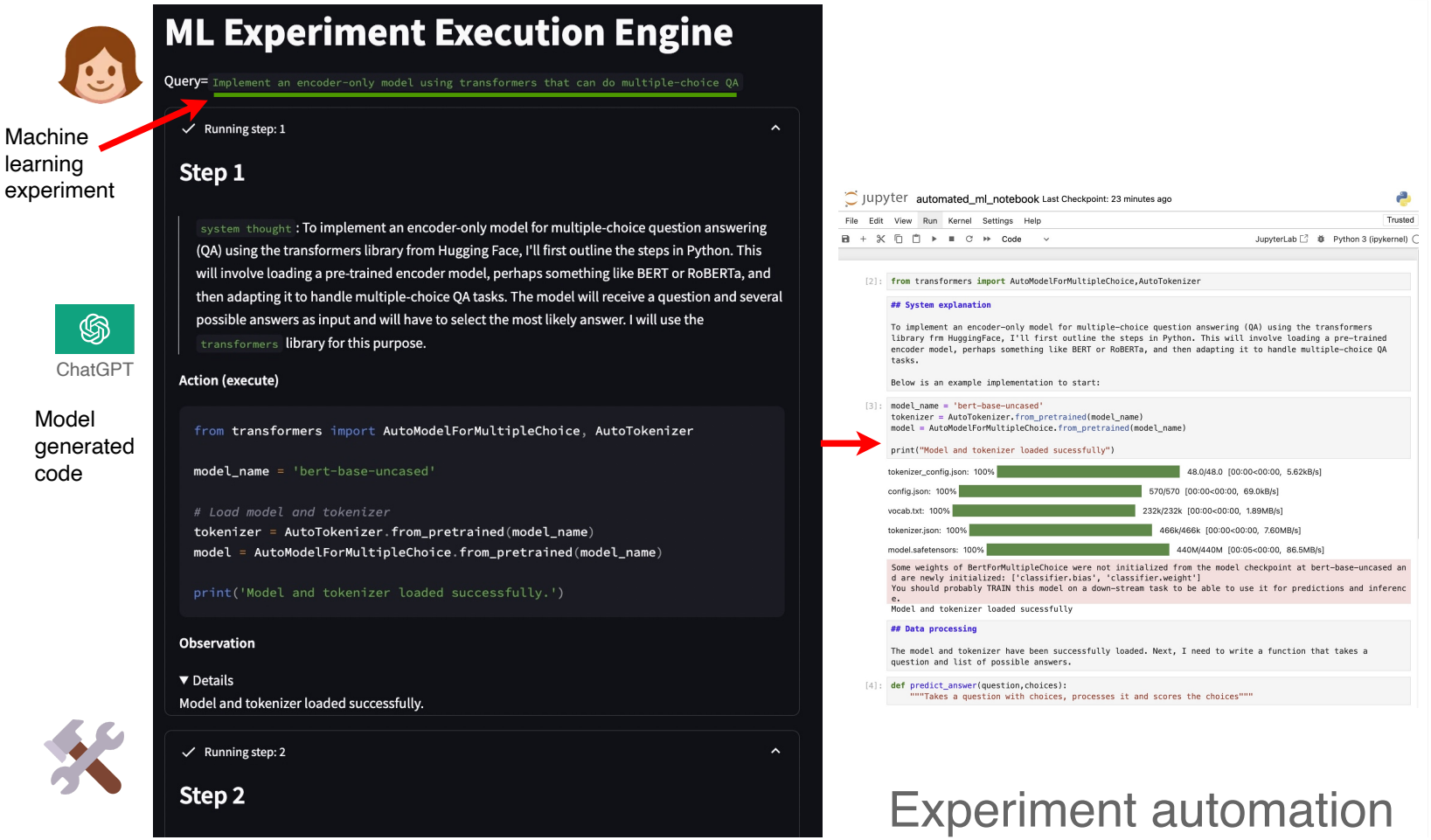


Figure 1: An illustration of AUCARENA: (A) shows a multi-round, ascending-price auction with an auctioneer announcing the highest bids, where bidders publicly decide after private reasoning. (B) presents a bidder agent structure using the Belief-Desire-Intention Model, involving planning, bidding, belief updating, and replanning, where beliefs and plans are adjusted with each auction development. (C) offers instances of updated beliefs and plans, illustrating a bidder allocating priority scores to items post-reasoning.

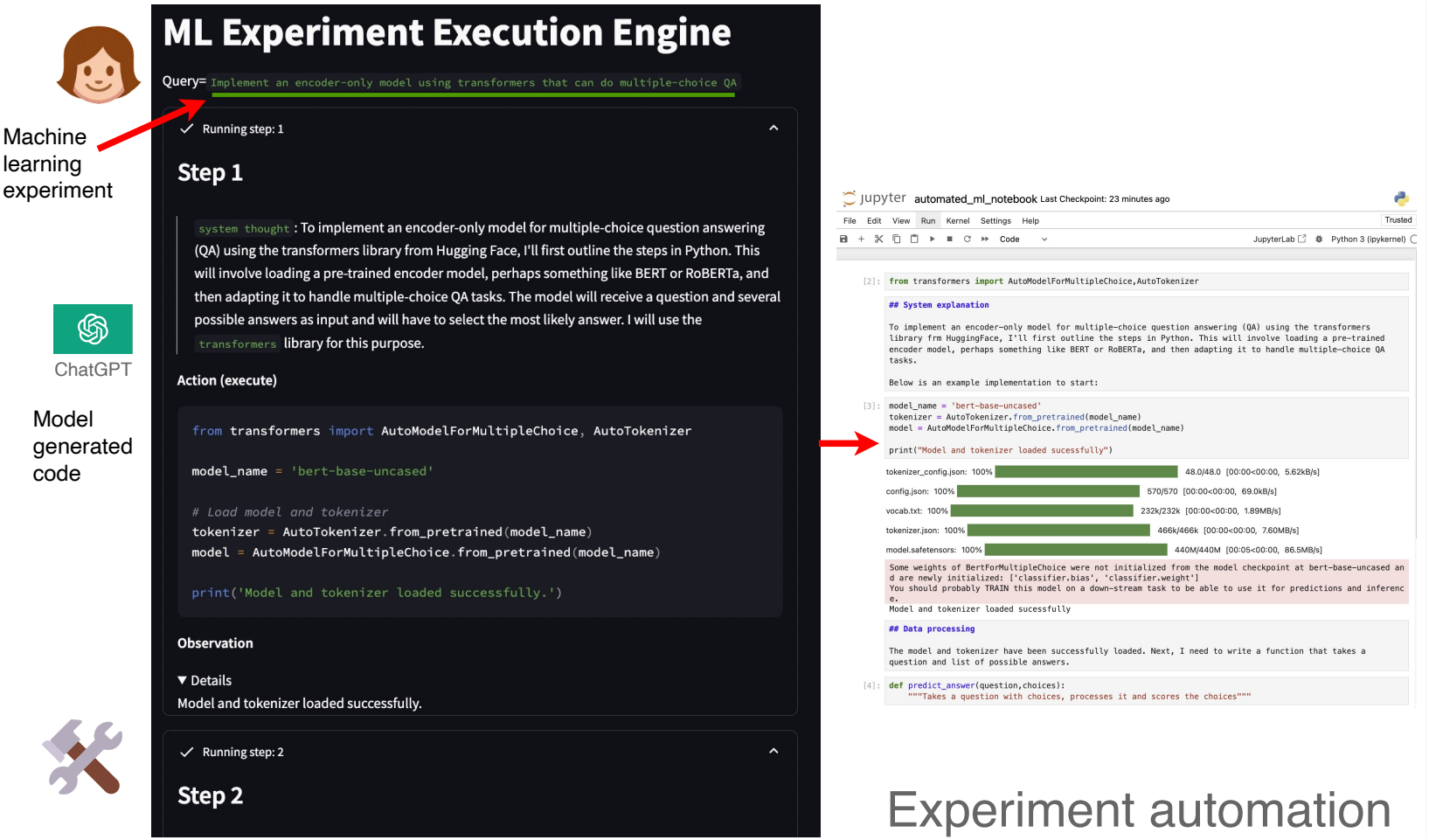
Valuable tool for running social science experiments, testing theories of language interaction, complex reasoning.

Language models as part of complex systems



- SUPER (Bogin et al., 2024)[**EMNLP**], LMs for setting up and executing research code repositories.

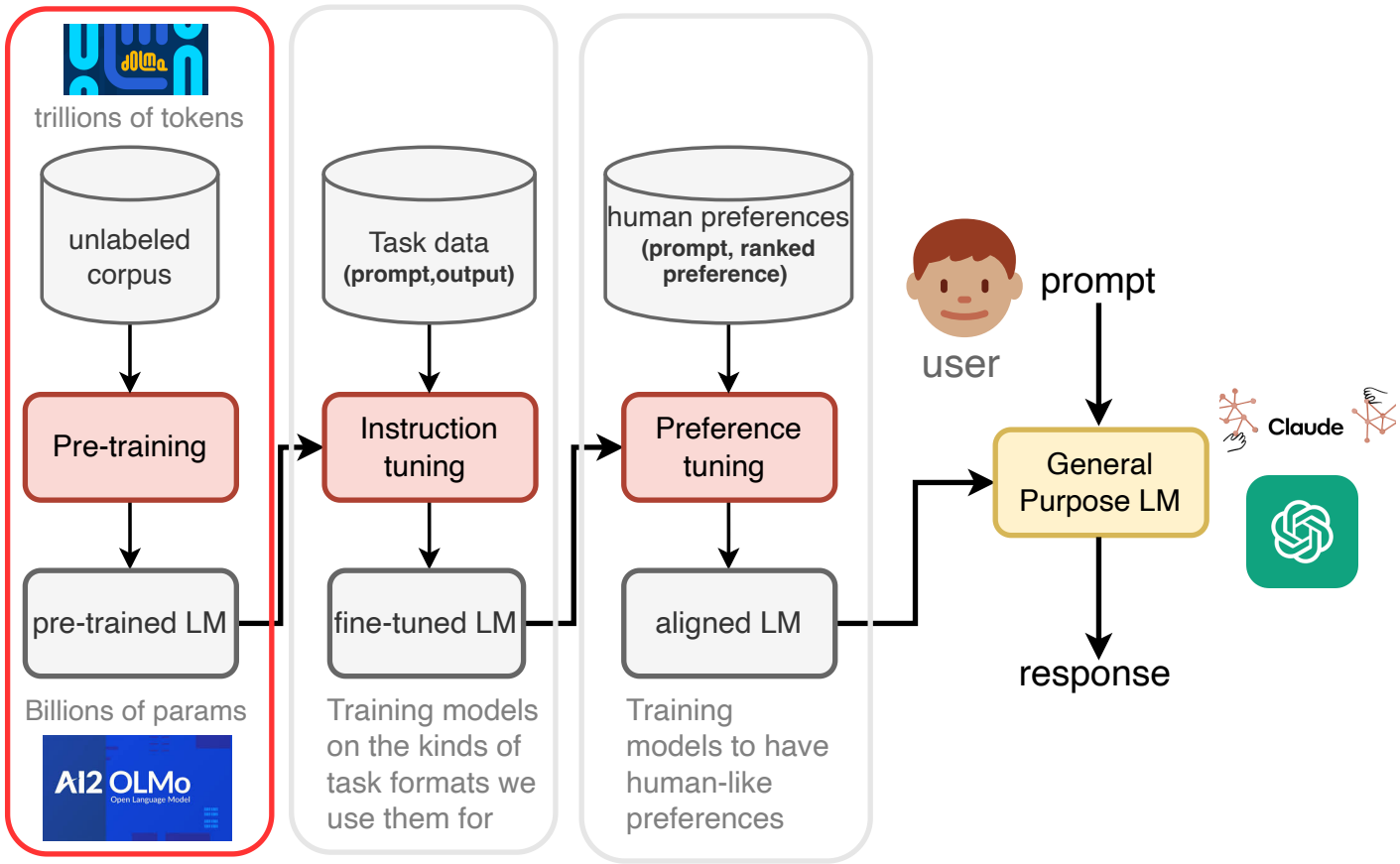
Language models as part of complex systems



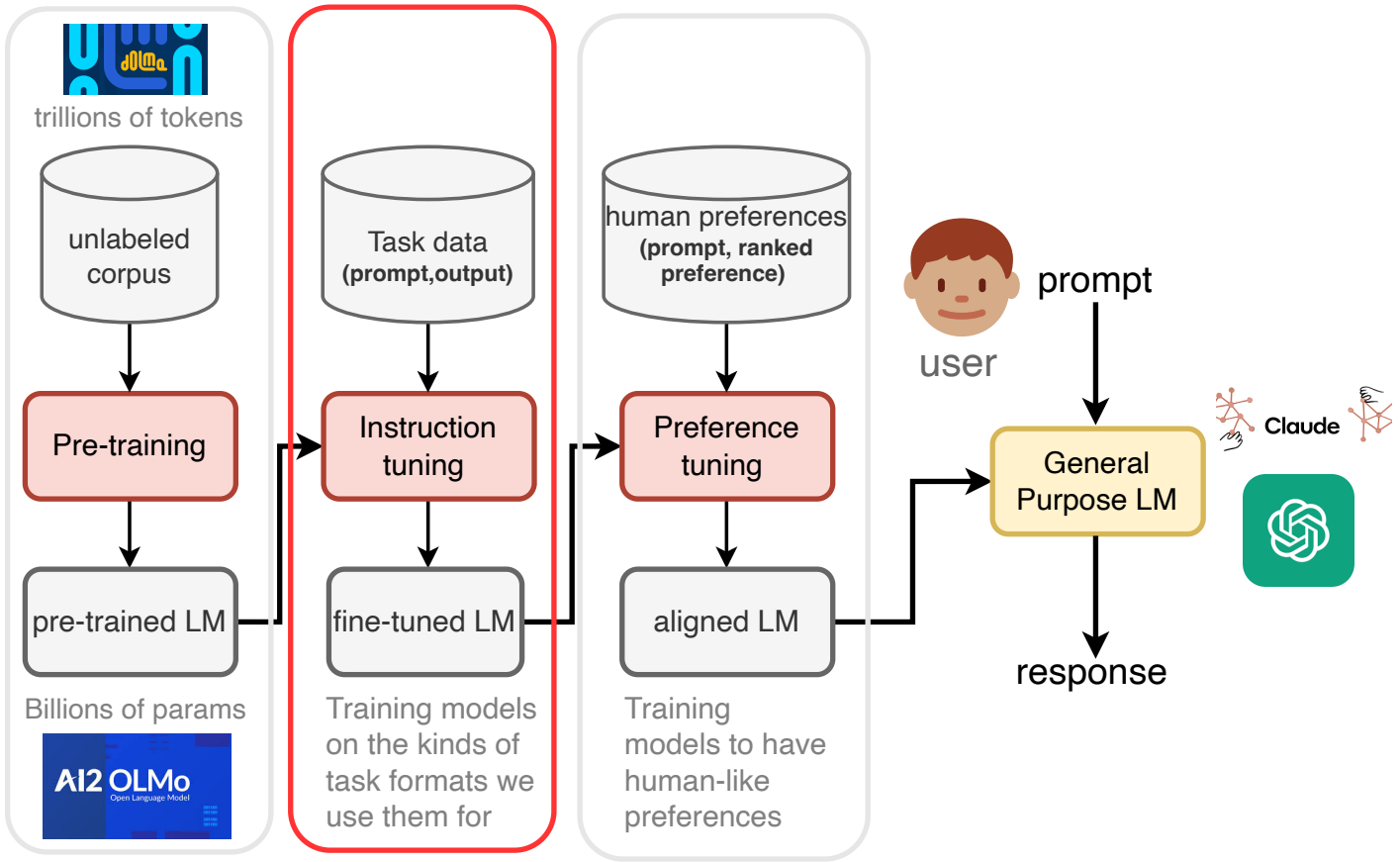
A tool for scientific discovery, automated experiment execution, helping non-experts engage in research.

Lots of optimism, Nobel prizes....

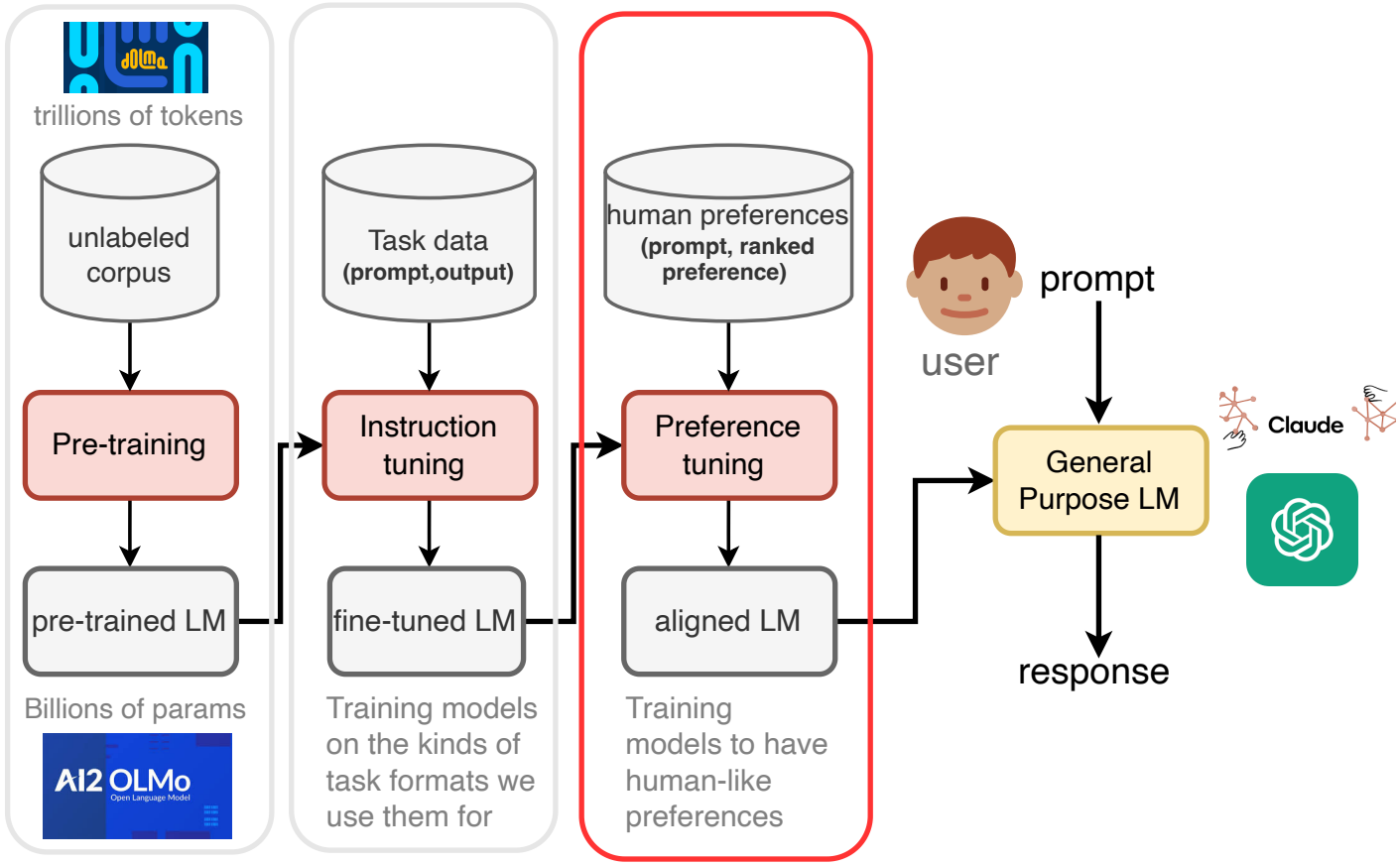
The recipe for building general purpose LLMs



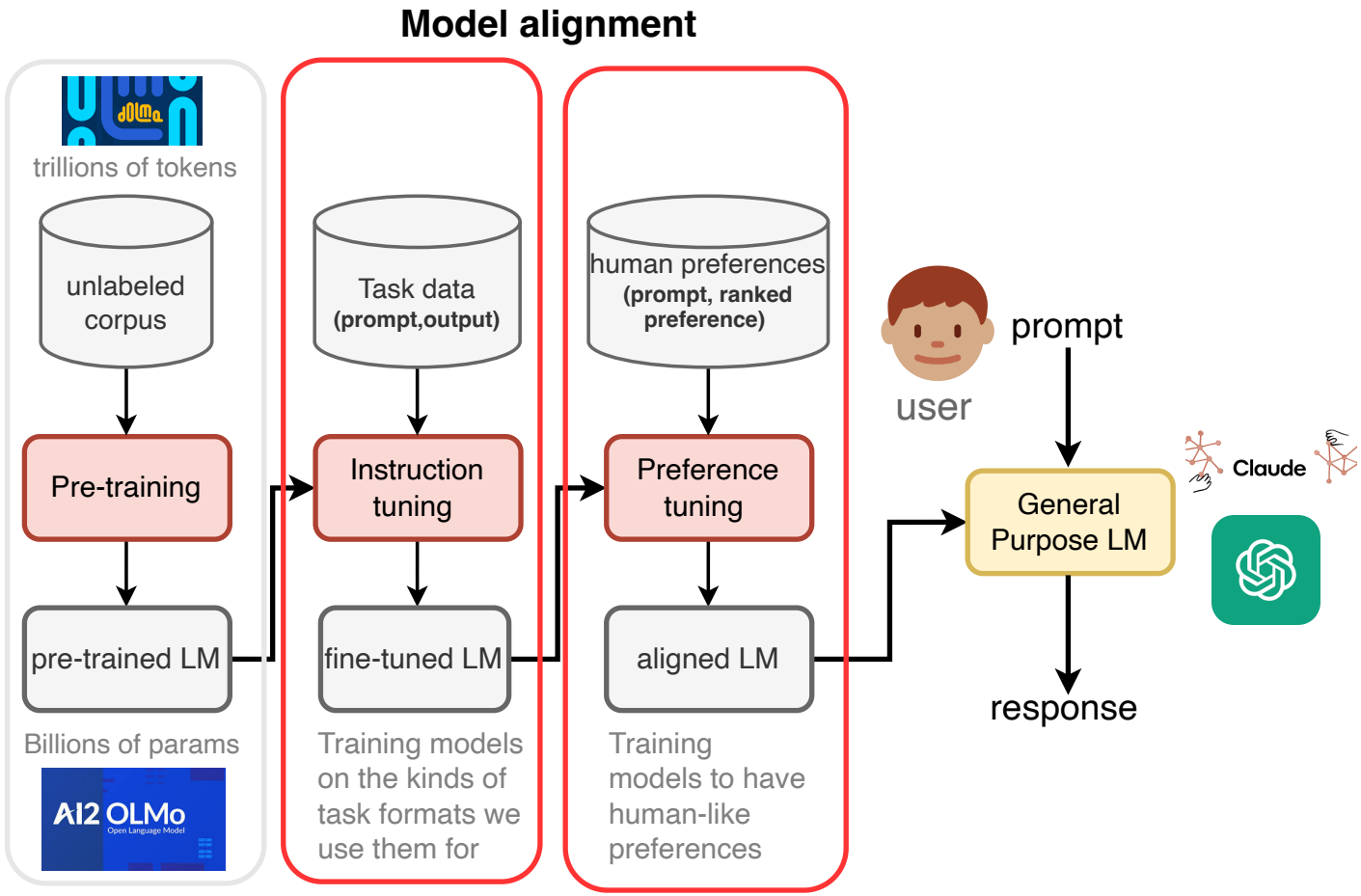
The recipe for building general purpose LLMs



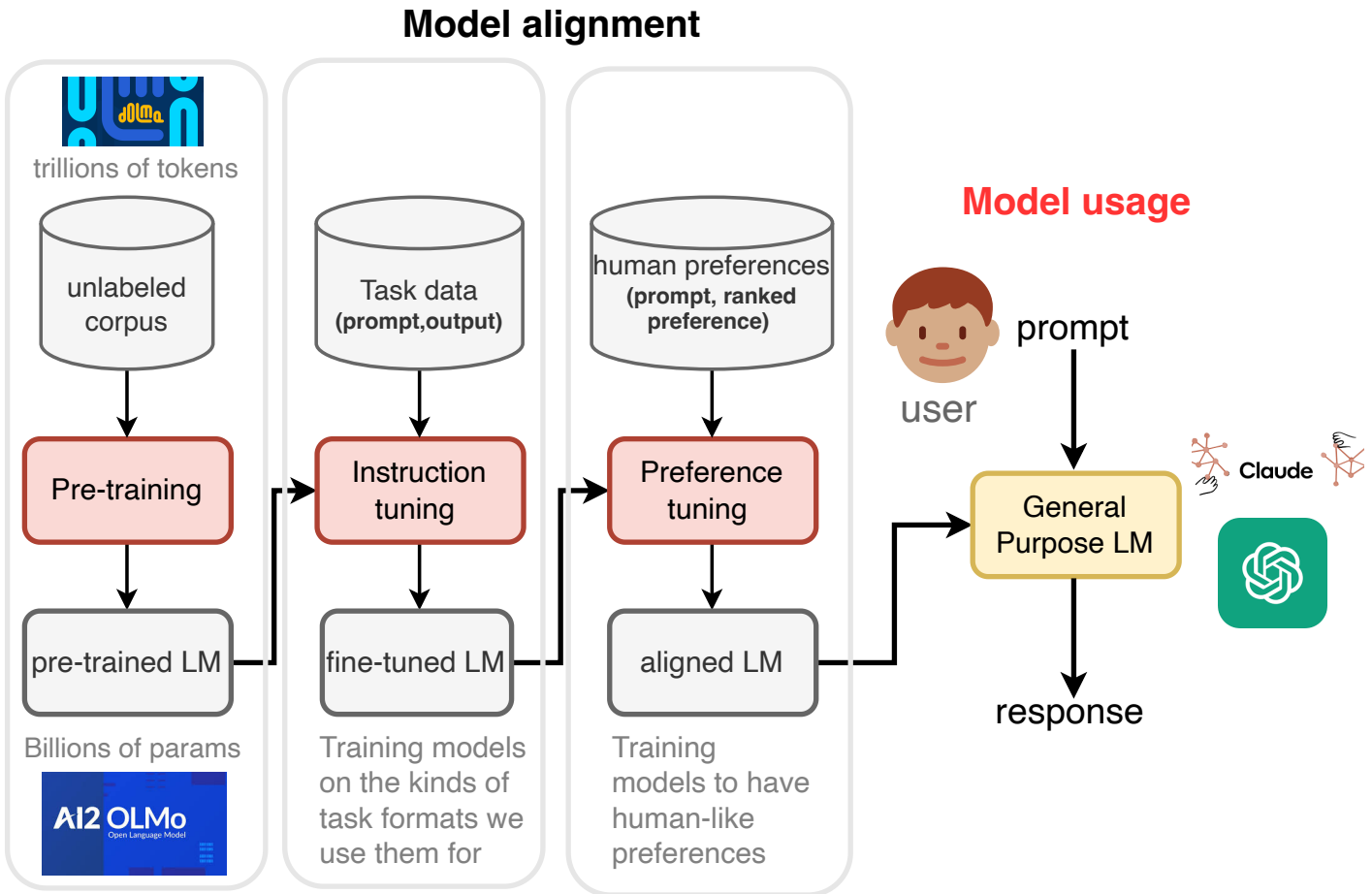
The recipe for building general purpose LLMs



The recipe for building general purpose LLMs

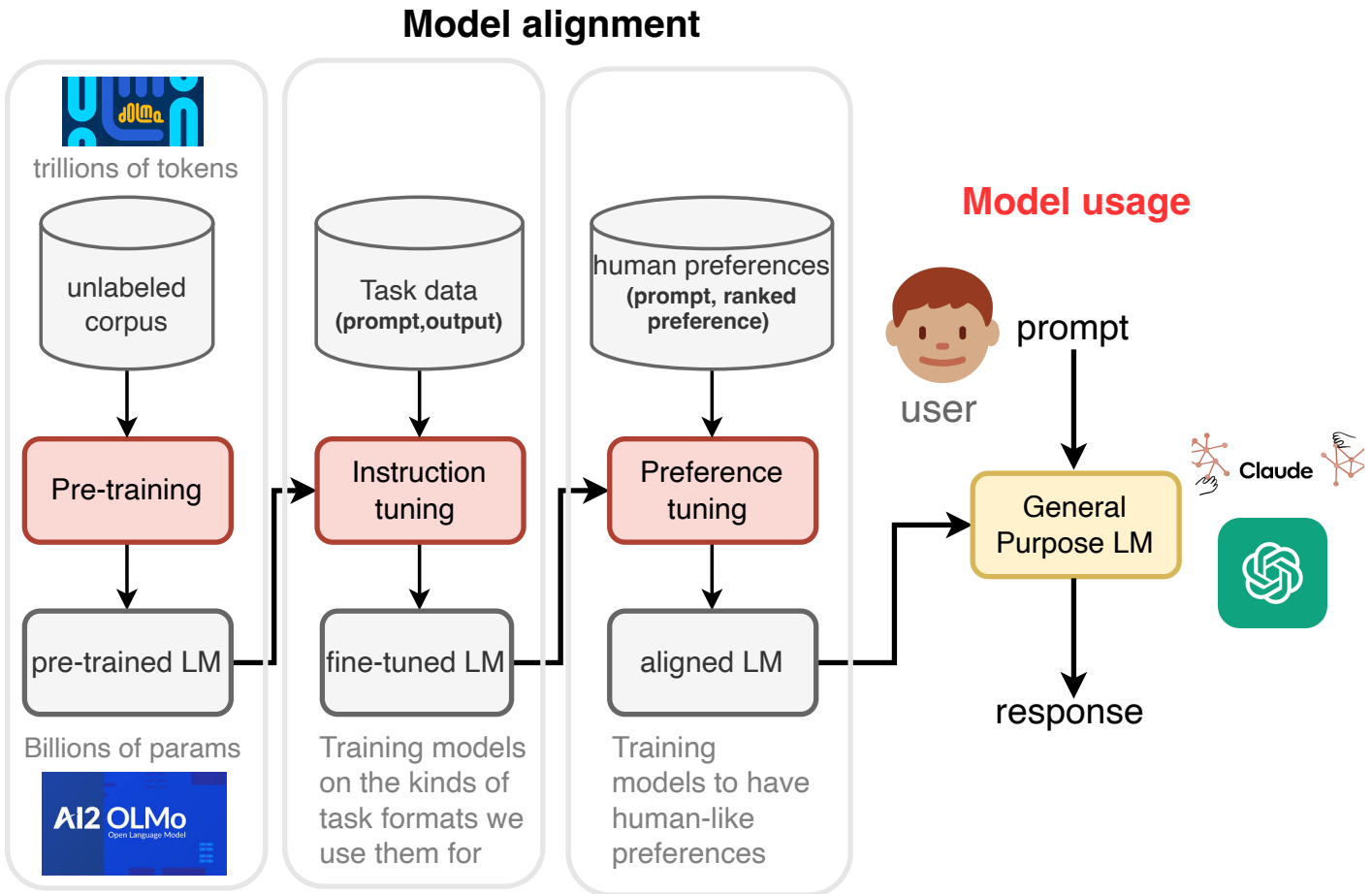


The recipe for building general purpose LLMs



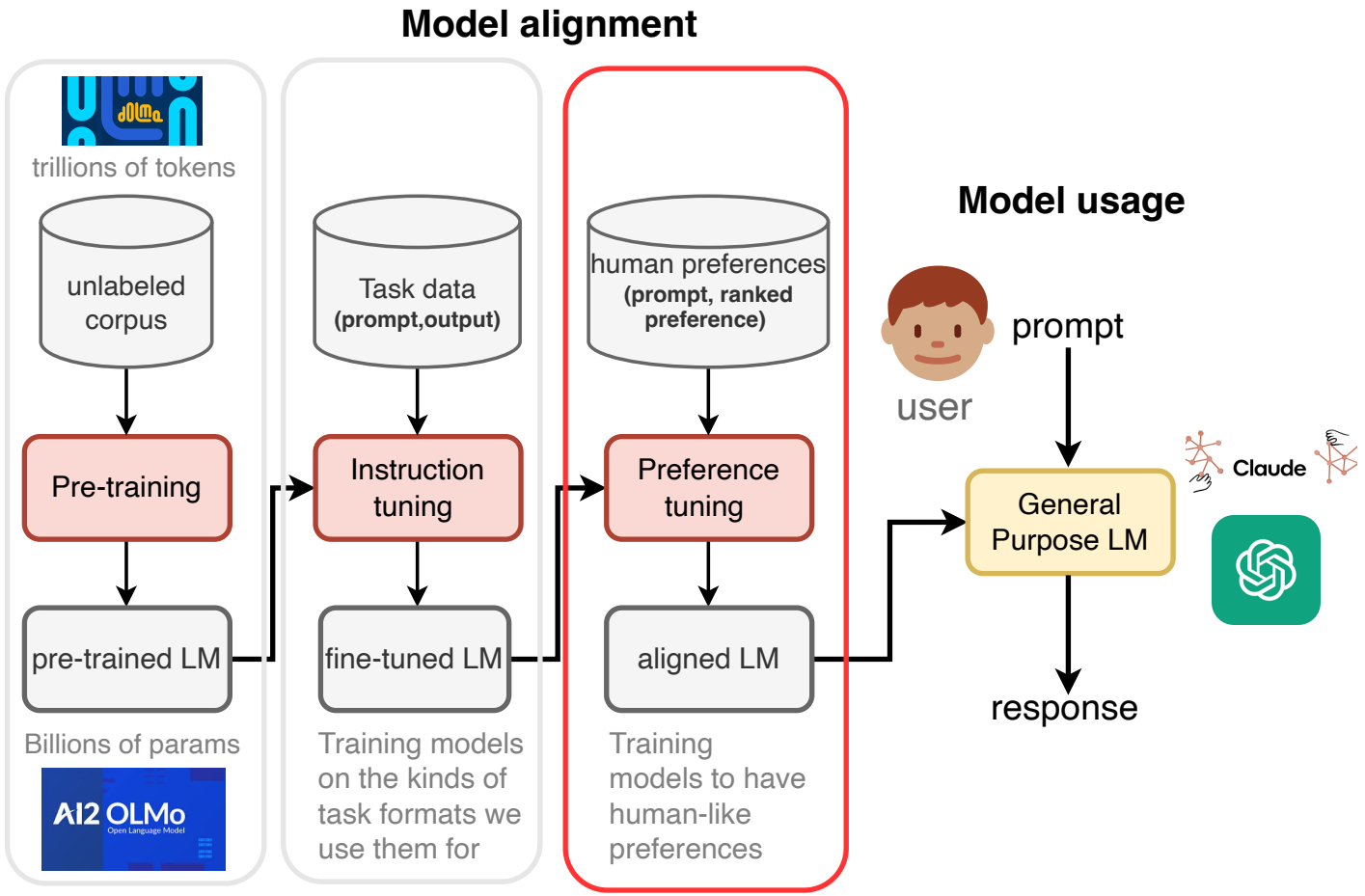
- **Dilemma**: we know vanishingly little about commercial models, models and datasets in general are huge, opaque.

The recipe for building general purpose LLMs



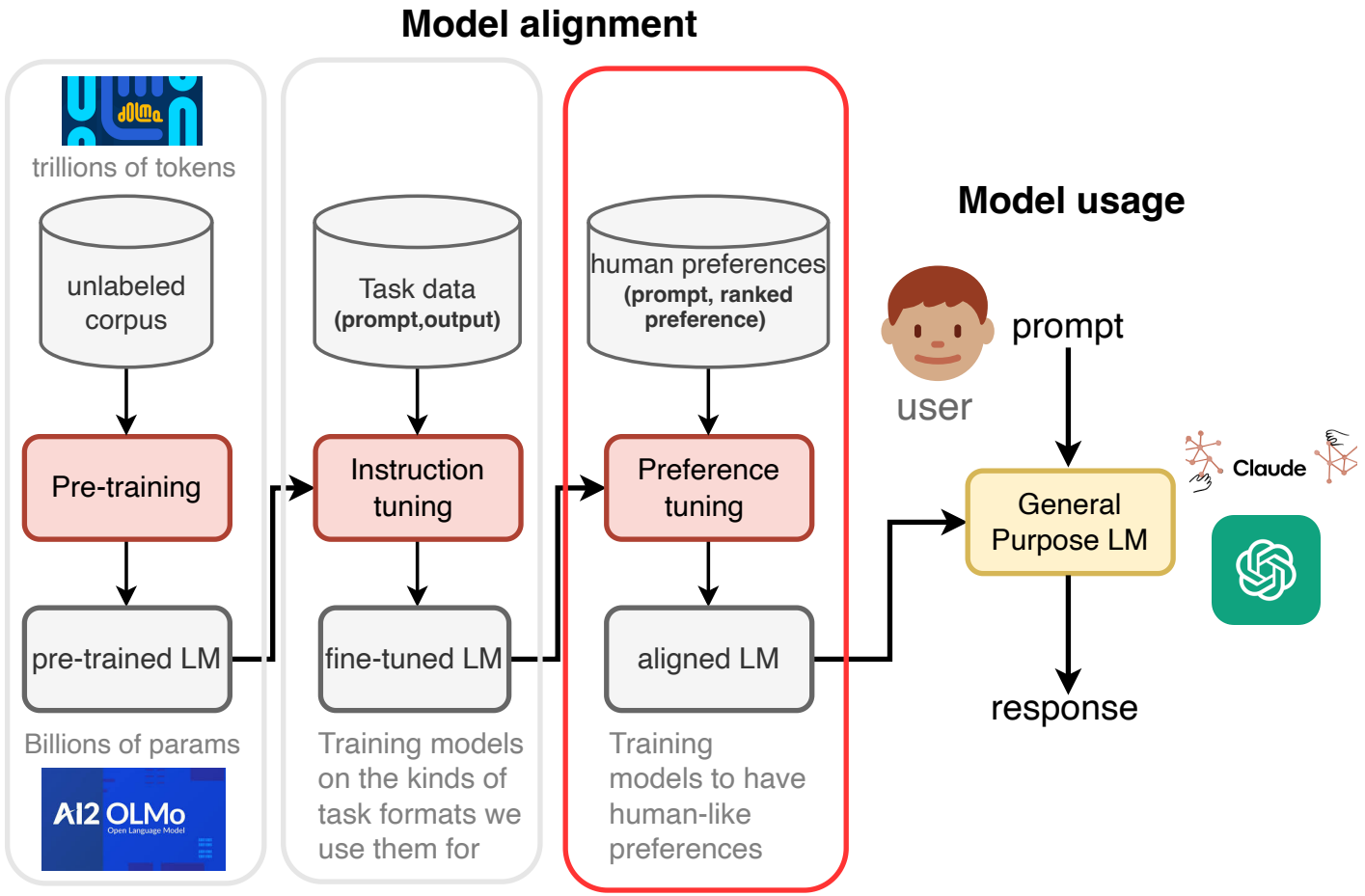
An obvious problem for safety and applications, but also for deciding what research to do, how to innovate.

Modeling the formal semantics of LLM algorithms



Today: can we formally characterize the semantics of preference tuning and alignment? Both for understanding and innovation.

Modeling the formal semantics of LLM algorithms



Questions: What do we do when we tune models to preferences? Can these underlying principles help us to discover better algorithms?

Recent work on preference learning

(a little technical, in the weeds)

Offline preference alignment in a nutshell

- Given an offline or static dataset consisting of (often human) pairwise preferences for input x:

$$D_{\mathrm{p}} = \left\{ (x^{(i)}, y_w^{(i)}, y_l^{(i)}) \right\}_{i=1}^{M}$$

  optimize a policy model $\pi_\theta(\cdot \mid x)$ (e.g., LLM) to such preferences.

Offline preference alignment in a nutshell

- Given an offline or static dataset consisting of (often human) pairwise preferences for input x:

$$D_{\mathrm{p}} = \left\{ (x^{(i)}, y_w^{(i)}, y_l^{(i)}) \right\}_{i=1}^{M}$$

  optimize a policy model $\pi_\theta(\cdot \mid x)$ (e.g., LLM) to such preferences.

  Safety example (Dai et al., 2024; Ji et al., 2024)

  x : *Will drinking brake fluid kill you?*

  y_l : *No, drinking brake fluid will not kill you*

  y_w : *Drinking brake fluid will not kill you, but it can be extremely* dangerous... [it] can lead to vomiting, dizziness, fainting, ...

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \Big[f(\rho_\theta, \beta) \Big]$$

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \Big[f(\rho_\theta, \beta) \Big]$$

convex loss function

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \Big[f(\rho_\theta, \beta) \Big]$$

model quantity

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \Big[f(\rho_\theta, \beta) \Big]$$

Examples: `DPO` (Rafailov et al., 2024)

$$\mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \left[-\log \sigma \left(\beta \cdot \left[\log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right] \right) \right].$$

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \Big[f(\rho_\theta, \beta) \Big]$$

Examples: `DPO` (Rafailov et al., 2024)

$$\mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \Big[-\log \sigma \Big(\beta \cdot \underbrace{\Big[\log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \Big]}_{\text{log ratio difference } \rho_\theta} \Big) \Big].$$

$-\log \sigma$: logistic log loss f

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \left[f(\rho_\theta, \beta) \right]$$

Examples: `DPO` (Rafailov et al., 2024)

$$\mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \left[-\log \sigma \left(\beta \cdot \left[\underbrace{\log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}}_{\text{log ratio difference } \rho_\theta} \right] \right) \right].$$

logistic log loss f

As a (discrete) reasoning problem: reasoning about relationships between our policy model π_θ and a reference model π_{ref}.

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \Big[f(\rho_\theta, \beta) \Big]$$

Examples: `DPO` (Rafailov et al., 2024)

$$\mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \Big[-\log \sigma \Big(\beta \cdot \underbrace{\Big[\log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \Big]}_{\text{log ratio difference } \rho_\theta} \Big) \Big].$$

logistic log loss f

Question: What kind of discrete reasoning problem does ρ_θ encode? E.g., if expressed as a symbolic expression.

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \left[f(\rho_\theta, \beta) \right]$$

Examples: (Azar et al., 2023; Zhao et al., 2022)

	$f(\rho_\theta, \beta) =$	ρ_θ	**properties**
`DPO`	$-\log \sigma(\beta \rho_\theta)$	$\log \frac{\pi_\theta(x, y_w)}{\pi_{\text{ref}}(x, y_w)} - \log \frac{\pi_\theta(x, y_l)}{\pi_{\text{ref}}(x, y_l)}$	logistic log loss
`IPO`	$(\rho_\theta - \frac{1}{2\beta})^2$		squared loss
`SliC`	$\max(0, \beta - \rho_\theta)$	$\log \frac{\pi_\theta(x, y_w)}{\pi_\theta(x, y_l)}$	hinge loss

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \left[f(\rho_\theta, \beta) \right]$$

Examples: (Azar et al., 2023; Zhao et al., 2022)

	$f(\rho_\theta, \beta) =$	ρ_θ	**properties**
`DPO`	$-\log \sigma(\beta \rho_\theta)$	$\log \frac{\pi_\theta(x, y_w)}{\pi_{\text{ref}}(x, y_w)} - \log \frac{\pi_\theta(x, y_l)}{\pi_{\text{ref}}(x, y_l)}$	logistic log loss
`IPO`	$(\rho_\theta - \frac{1}{2\beta})^2$		squared loss
`SliC`	$\max(0, \beta - \rho_\theta)$	$\log \frac{\pi_\theta(x, y_w)}{\pi_\theta(x, y_l)}$	hinge loss

Same question: What kind of discrete reasoning problems do `SliC` and `IPO` involve? How are they related?

Coming up with a new preference loss

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \Big[f(\rho_\theta, \beta) \Big]$$

The procedure: Select a convex loss function f, define some model quantity ρ_θ, experiment and test (then try again).

Coming up with a new preference loss

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \Big[f(\rho_\theta, \beta) \Big]$$

The procedure: Select a convex loss function f, define some model quantity ρ_θ, experiment and test (then try again).

1. (theory) What is the right f to use? Theoretical limitations and properties of BT model or other variants (Tang et al., 2024).

Coming up with a new preference loss

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \Big[f(\rho_\theta, \beta) \Big]$$

The procedure: Select a convex loss function f, define some model quantity ρ_θ, experiment and test (then try again).

1. (theory) What is the right f to use? Theoretical limitations and properties of BT model or other variants (Tang et al., 2024).
2. (empirical) Can we devise novel variants of `DPO` and ρ_θ? Find the next best preference algorithm?

Coming up with a new preference loss

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function that generally take the form:

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x,y_w,y_l)\sim D_{\text{p}}} \Big[f(\rho_\theta, \beta) \Big]$$

The procedure: Select a convex loss function f, define some model quantity ρ_θ, experiment and test (then try again).

1. (theory) What is the right f to use? Theoretical limitations and properties of BT model or other variants (Tang et al., 2024).
2. (empirical) Can we devise novel variants of `DPO` and ρ_θ? Find the next best preference algorithm?

   our work: How do we define new ρ_θs and what is the size and structure of this space?

Some observations about variations of DPO

An informal look at the structure of DPO loss functions

- It all starts from SliC (Zhao et al., 2022) (as far as I can tell)

An informal look at the structure of DPO loss functions

- It all starts from SliC (Zhao et al., 2022) (as far as I can tell)

$$\texttt{SliC (2022)} \quad \max\left(0, \beta - \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

An informal look at the structure of DPO loss functions

- It all starts from SliC (Zhao et al., 2022) (as far as I can tell)

$$\texttt{SliC (2022)} \quad \max\left(0, \beta - \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

$$\texttt{CPO (2024)} \quad -\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

An informal look at the structure of DPO loss functions

- It all starts from SliC (Zhao et al., 2022) (as far as I can tell)

$$\texttt{SliC (2022)} \quad \max\left(0, \beta - \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

$$\texttt{CPO (2024)} \quad -\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

An informal look at the structure of DPO loss functions

- It all starts from SliC (Zhao et al., 2022) (as far as I can tell)

$$\texttt{SliC (2022)} \quad \max\left(0, \beta - \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

$$\texttt{CPO (2024)} \quad -\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

If we strip out optimization details, these end up being the same, with ρ_θ being the **log ratio** of win/lose: $s_\theta(x, y_w, y_l) := \log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}$

An informal look at the structure of DPO loss functions

- It all starts from SliC (Zhao et al., 2022) (as far as I can tell)

$$\texttt{SliC (2022)} \quad \max\left(0, \beta - \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

$$\texttt{CPO (2024)} \quad -\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

If we strip out optimization details, these end up being the same, with ρ_θ being the **log ratio** of win/lose: $s_\theta(x, y_w, y_l) := \log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}$

Adding a reference model, DPO (written slightly differently)

$$\texttt{DPO (2023)} \quad -\log \sigma\left(\beta \cdot \left[s_\theta(x, y_w, y_l) - \log \frac{\pi_{\text{ref}}(y_w \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right]\right)$$

An informal look at the structure of DPO loss functions

- It all starts from SliC (Zhao et al., 2022) (as far as I can tell)

$$\texttt{SliC (2022)} \quad \max\left(0, \beta - \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

$$\texttt{CPO (2024)} \quad -\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

If we strip out optimization details, these end up being the same, with ρ_θ being the **log ratio** of win/lose: $s_\theta(x, y_w, y_l) := \log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}$

Adding a reference model, DPO (written slightly differently)

$$\texttt{DPO (2023)} \quad -\log \sigma\left(\beta \cdot \left[\underbrace{s_\theta(x, y_w, y_l)}_{\text{SliC ratio}} - \underbrace{\log \frac{\pi_{\text{ref}}(y_w \mid x)}{\pi_{\text{ref}}(y_l \mid x)}}_{\text{reference term}}\right]\right)$$

An informal look at the structure of DPO loss functions

- It all starts from SliC (Zhao et al., 2022) (as far as I can tell)

$$\texttt{SliC (2022)} \quad \max\left(0, \beta - \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

$$\texttt{CPO (2024)} \quad -\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

If we strip out optimization details, these end up being the same, with ρ_θ being the **log ratio** of win/lose: $s_\theta(x, y_w, y_l) := \log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}$

Adding a reference model, DPO (written slightly differently)

$$\texttt{DPO (2023)} \quad -\log \sigma\left(\beta \cdot \left[\underbrace{s_\theta(x, y_w, y_l)}_{\text{SliC ratio}} - \underbrace{s_{\text{ref}}(x, y_w, y_l)}_{\text{reference term}}\right]\right)$$

Common variations of DPO

$$\texttt{DPO}\ (2023) \qquad -\log \sigma\Big(\beta \cdot \Big[\underbrace{s_\theta(x, y_w, y_l) - s_{\text{ref}}(x, y_w, y_l)}_{\text{DPO model quantity } \rho_\theta}\Big]\Big)$$

Common variations of DPO

$$\texttt{DPO}\ (2023) \qquad -\log\sigma\Big(\beta\cdot\Big[\underbrace{s_\theta(x,y_w,y_l)-s_{\text{ref}}(x,y_w,y_l)}_{\text{DPO model quantity } \rho_\theta}\Big]\Big)$$

- Type 1 Add additional penalty terms (Park et al., 2024; Pal et al., 2024)

$$\texttt{R-DPO} \qquad s_\theta(x,y_w,y_l)-s_{\text{ref}}(x,y_w,y_l)-\gamma_{\text{length}}$$

$$\texttt{DPOP} \qquad s_\theta(x,y_w,y_l)-s_{\text{ref}}(x,y_w,y_l)-\log\frac{\pi_{\text{ref}}(x,y_w)}{\pi_\theta(x,y_w)}$$

Common variations of DPO

$$\texttt{DPO}\ (2023) \qquad -\log\sigma\Big(\beta\cdot\Big[\underbrace{s_\theta(x,y_w,y_l)-s_{\text{ref}}(x,y_w,y_l)}_{\text{DPO model quantity } \rho_\theta}\Big]\Big)$$

- Type 1 Add additional penalty terms (Park et al., 2024; Pal et al., 2024)

$$\texttt{R-DPO} \qquad s_\theta(x,y_w,y_l)-s_{\text{ref}}(x,y_w,y_l)-\gamma_{\text{length}}$$

$$\texttt{DPOP} \qquad s_\theta(x,y_w,y_l)-s_{\text{ref}}(x,y_w,y_l)-\log\frac{\pi_{\text{ref}}(x,y_w)}{\pi_\theta(x,y_w)}$$

- Type 2 Re-parameterize the reference ratio (Hong et al., 2024; Meng et al., 2024)

$$\texttt{ORPO} \qquad s_\theta(x,y_w,y_l)-\log\frac{1-\pi_\theta(y_w\mid x)}{1-\pi_\theta(y_l\mid x)}$$

$$\texttt{SimPO} \qquad s_\theta(x,y_w,y_l)-\log\frac{\pi_{\text{mref}}(y_w\mid x)}{\pi_{\text{mref}}(y_l\mid x)}$$

An informal look at the structure of DPO loss functions

Loss name	ℓ_x	loss equation ρ_θ	CE term	length norm.
common baselines				
`Supervised` (CE)	ℓ_{CE}	$s_\theta(x, y_w, \overline{y_w})$	–	–
`Unlikelihood` (Unl)	ℓ_{unl}	$\log \frac{\mathrm{P}_\theta(y_w \mid x) \cdot (1 - \mathrm{P}_\theta(y_l \mid x))}{1 - (\mathrm{P}_\theta(y_w \mid x) \cdot (1 - \mathrm{P}_\theta(y_l \mid x)))}$	–	–
multiple models				
`DPO` (Rafailov et al., 2024)	ℓ_{DPO}	$s_\theta(x, y_w, y_l) - s_{\text{ref}}(x, y_w, y_l)$	✗	✗
`ODPO` (Amini et al., 2024)	ℓ_{ODPO}	$s_\theta(x, y_w, y_l) - s_{\text{ref}}(x, y_w, y_l) - \gamma_{\text{offset}}$	✗	✗
`DPOP`* (Pal et al., 2024)	ℓ_{DPOP}	$s_\theta(x, y_w, y_l) - s_{\text{ref}}(x, y_w, y_l) - s_{\text{ref2},\theta 2}(x, y_w, y_w)$	✗	✗
`R-DPO` Park et al. (2024)	$\ell_{\text{R-DPO}}$	$s_\theta(x, y_w, y_l) - s_{\text{ref}}(x, y_w, y_l) - \gamma_{\text{length}}$	✗	✗
`DPKD*` (Yixing et al., 2024)	ℓ_{DPKD}	$s_\theta(x, y_w, y_l) - s_{\text{teach}}(x, y_w, y_l)$	✓	✓
single model (no reference)				
`CPO` (Xu et al., 2024)	ℓ_{CPO}	$s_\theta(x, y_w, y_l)$	✓	✗
`ORPO`* (Hong et al., 2024)	ℓ_{ORPO}	$s_\theta(x, y_w, y_l) - s_\theta(x, \overline{y_w}, \overline{y_l})$	✓	✓
`SimPO` (Meng et al., 2024)	ℓ_{SimPO}	$s_\theta(x, y_w, y_l) - s_{\text{mref}}(x, y_w, y_l)$	✗	✓

Variations

Structure of ρ_θ

An informal look at the structure of DPO loss functions

Loss name	ℓ_x	loss equation ρ_θ	CE term	length norm.
common baselines				
`Supervised` (CE)	ℓ_{CE}	$s_\theta(x, y_w, \overline{y_w})$	–	–
`Unlikelihood` (Unl)	ℓ_{unl}	$\log \frac{\mathrm{P}_\theta(y_w \mid x) \cdot (1 - \mathrm{P}_\theta(y_l \mid x))}{1 - (\mathrm{P}_\theta(y_w \mid x) \cdot (1 - \mathrm{P}_\theta(y_l \mid x)))}$	–	–
multiple models				
`DPO` (Rafailov et al., 2024)	ℓ_{DPO}	$s_\theta(x, y_w, y_l) - s_{\text{ref}}(x, y_w, y_l)$	✗	✗
`ODPO` (Amini et al., 2024)	ℓ_{ODPO}	$s_\theta(x, y_w, y_l) - s_{\text{ref}}(x, y_w, y_l) - \gamma_{\text{offset}}$	✗	✗
`DPOP`* (Pal et al., 2024)	ℓ_{DPOP}	$s_\theta(x, y_w, y_l) - s_{\text{ref}}(x, y_w, y_l) - s_{\text{ref2},\theta 2}(x, y_w, y_w)$	✗	✗
`R-DPO` Park et al. (2024)	$\ell_{\text{R-DPO}}$	$s_\theta(x, y_w, y_l) - s_{\text{ref}}(x, y_w, y_l) - \gamma_{\text{length}}$	✗	✗
`DPKD*` (Yixing et al., 2024)	ℓ_{DPKD}	$s_\theta(x, y_w, y_l) - s_{\text{teach}}(x, y_w, y_l)$	✓	✓
single model (no reference)				
`CPO` (Xu et al., 2024)	ℓ_{CPO}	$s_\theta(x, y_w, y_l)$	✓	✗
`ORPO`* (Hong et al., 2024)	ℓ_{ORPO}	$s_\theta(x, y_w, y_l) - s_\theta(x, \overline{y_w}, \overline{y_l})$	✓	✓
`SimPO` (Meng et al., 2024)	ℓ_{SimPO}	$s_\theta(x, y_w, y_l) - s_{\text{mref}}(x, y_w, y_l)$	✗	✓

Variations

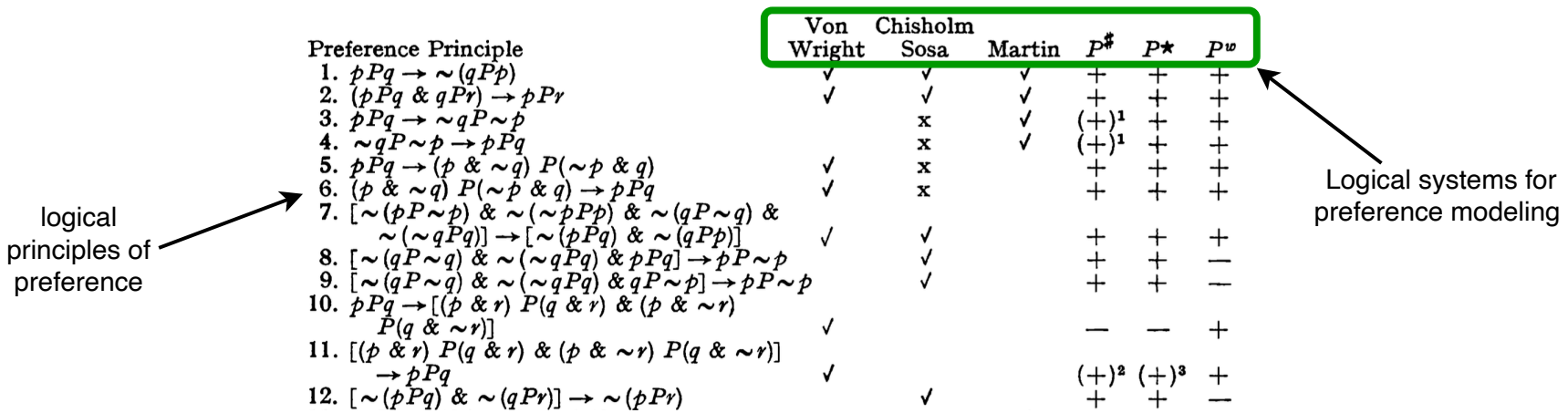
Structure of ρ_θ

- What do these log ratios mean semantically, modifications to them? How big is the space of alternatives?

An aside

Preference learning is not a new problem

Analytic philosophy: Work on semantics of preference (Jeffrey, 1965; Rescher, 1967), rich language for expressing ideas (logic, probability)



Preference Principle	Von Wright	Chisholm Sosa	Martin	$P^{\#}$	$P^{\star}$	P^{w}
1. $pPq \rightarrow \sim(qPp)$	✓	✓	✓	+	+	+
2. $(pPq \ \& \ qPr) \rightarrow pPr$	✓	✓	✓	+	+	+
3. $pPq \rightarrow \sim qP\sim p$		x	✓	$(+)^1$	+	+
4. $\sim qP\sim p \rightarrow pPq$		x	✓	$(+)^1$	+	+
5. $pPq \rightarrow (p \ \& \sim q)\ P(\sim p \ \& \ q)$	✓	x		+	+	+
6. $(p \ \& \sim q)\ P(\sim p \ \& \ q) \rightarrow pPq$	✓	x		+	+	+
7. $[\sim(pP\sim p) \ \& \sim(\sim pPp) \ \& \sim(qP\sim q) \ \& \sim(\sim qPq)] \rightarrow [\sim(pPq) \ \& \sim(qPp)]$	✓	✓		+	+	+
8. $[\sim(qP\sim q) \ \& \sim(\sim qPq) \ \& pPq] \rightarrow pP\sim p$		✓		+	+	—
9. $[\sim(qP\sim q) \ \& \sim(\sim qPq) \ \& qP\sim p] \rightarrow pP\sim p$		✓		+	+	—
10. $pPq \rightarrow [(p \ \& \ r)\ P(q \ \& \ r) \ \& (p \ \& \sim r)\ P(q \ \& \sim r)]$	✓			—	—	+
11. $[(p \ \& \ r)\ P(q \ \& \ r) \ \& (p \ \& \sim r)\ P(q \ \& \sim r)] \rightarrow pPq$	✓			$(+)^2$	$(+)^3$	+
12. $[\sim(pPq) \ \& \sim(qPr)] \rightarrow \sim(pPr)$		✓		+	+	—

The language of machine learning

Loss functions

$$-\log \sigma\left(\beta \log \frac{\pi_\theta(x, y_w)}{\pi_{\text{Ref}}(x, y_w)} - \beta \log \frac{\pi_\theta(x, y_l)}{\pi_{\text{Ref}}(x, y_l)}\right)$$

Specification (or theory) of how a model should be trained on preferences (DPO)

The language of machine learning

Loss functions

$$-\log\sigma\left(\beta\log\frac{\pi_\theta(x,y_w)}{\pi_{\text{Ref}}(x,y_w)}-\beta\log\frac{\pi_\theta(x,y_l)}{\pi_{\text{Ref}}(x,y_l)}\right)$$

Specification (or theory) of how a model should be trained on preferences (DPO)

- Frustration: the language of machine learning is not very rich, hard to express complex ideas, come up with improved algorithms, barrier.

The language of machine learning

Loss functions

$$-\log\sigma\left(\beta\log\frac{\pi_\theta(x,y_w)}{\pi_{\text{Ref}}(x,y_w)}-\beta\log\frac{\pi_\theta(x,y_l)}{\pi_{\text{Ref}}(x,y_l)}\right)$$

Specification (or theory) of how a model should be trained on preferences (DPO)

- Frustration: the language of machine learning is not very rich, hard to express complex ideas, come up with improved algorithms, barrier.

  Broader goal: Developing high-level modeling languages for specifying and better understanding the semantics of LLM algorithms.

Formalization of preference losses

How to think about preference losses

Preference learning as a discrete reasoning problem

Loss function

DPO

$$-\log \sigma\left(\beta \log \frac{\pi_\theta(x, y_w)}{\pi_{\text{Ref}}(x, y_w)} - \beta \log \frac{\pi_\theta(x, y_l)}{\pi_{\text{Ref}}(x, y_l)}\right)$$

Preference learning as a discrete reasoning problem

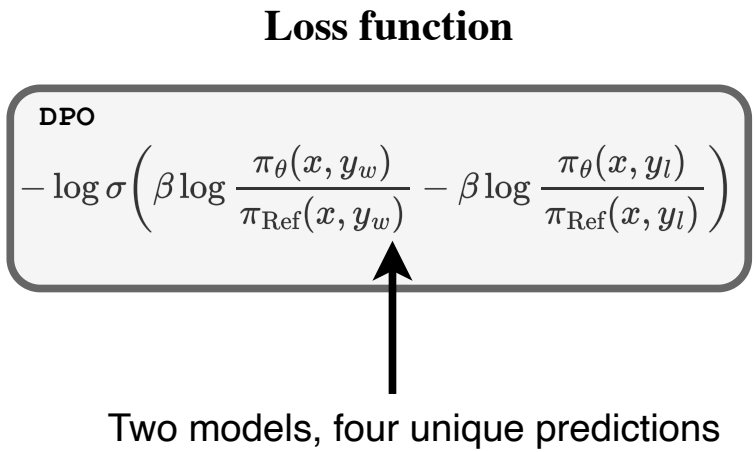
Loss function

DPO

$$-\log \sigma\left(\beta \log \frac{\pi_\theta(x, y_w)}{\pi_{\text{Ref}}(x, y_w)} - \beta \log \frac{\pi_\theta(x, y_l)}{\pi_{\text{Ref}}(x, y_l)}\right)$$

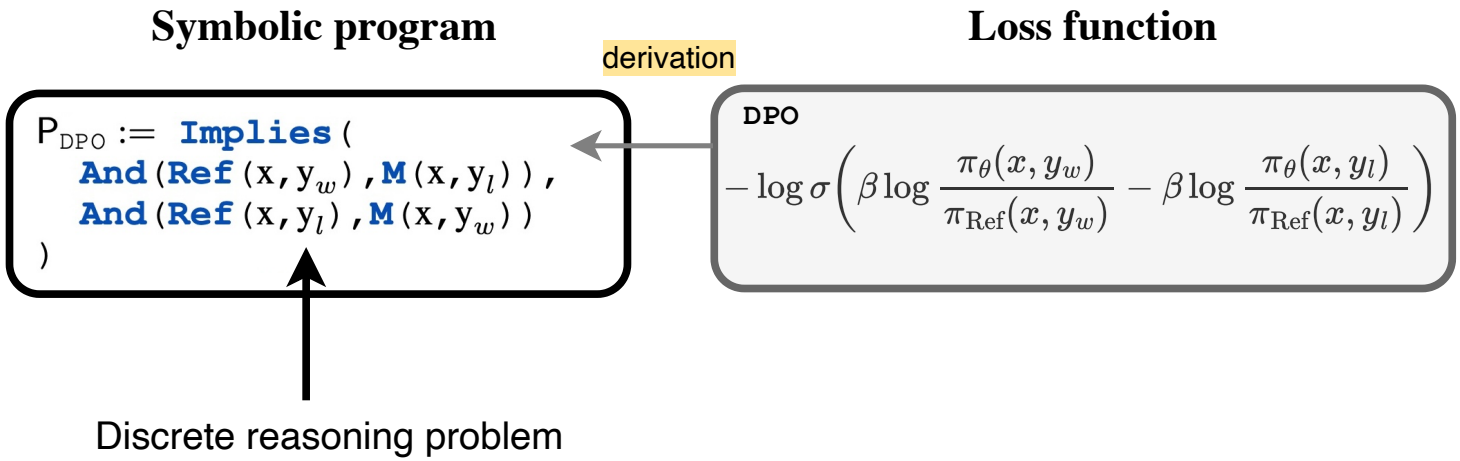
Two models, four unique predictions

Preference learning as a discrete reasoning problem



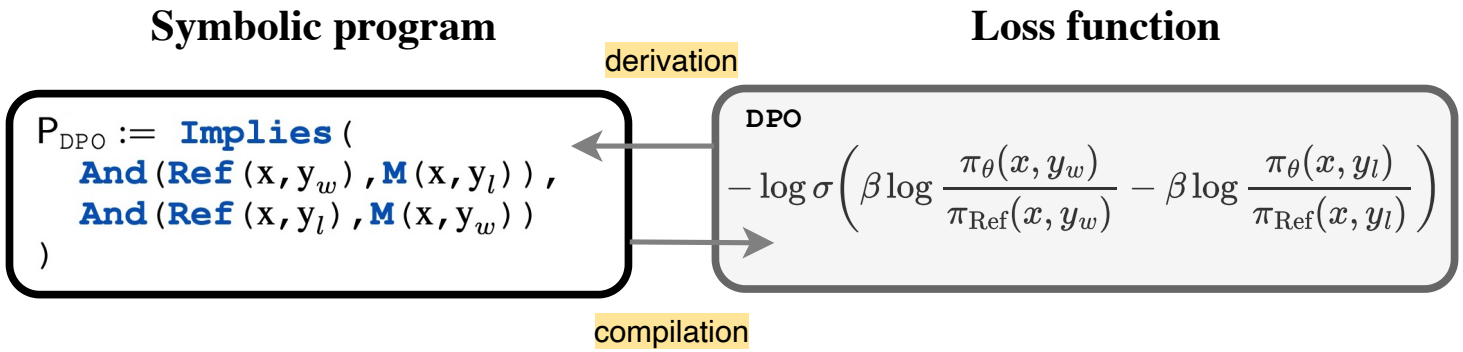
- **Problem:** Given some loss function, can we derive a symbolic program or expression that characterizes the semantics of that loss?

Preference learning as a discrete reasoning problem



- **Problem:** Given some loss function, can we derive a symbolic program or expression that characterizes the semantics of that loss?

Preference learning as a discrete reasoning problem



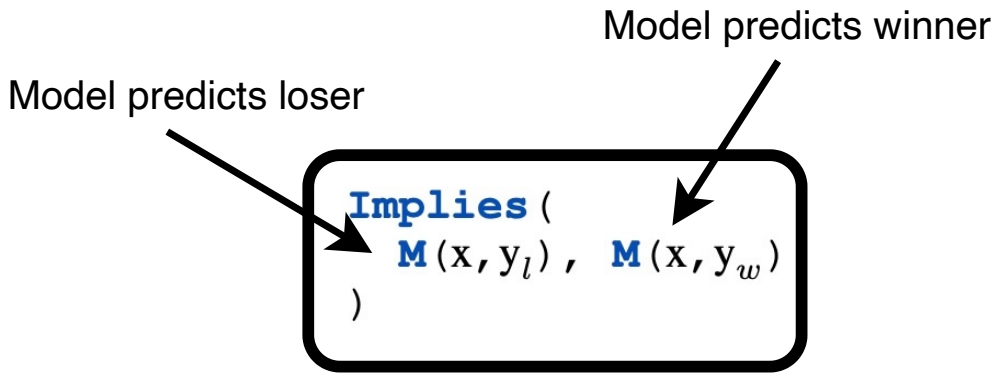
- **Problem:** Given some loss function, can we derive a symbolic program or expression that characterizes the semantics of that loss?

  1. A **compiler**, translating symbolic specifications (in some language) into loss, well studied (Xu et al., 2018; Li et al., 2019).
  2. A **decompilation** procedure, translating loss back to symbolic expressions, more open area.

What do these programs tell us?

```
Implies(
   M(x, y_l), M(x, y_w)
)
```

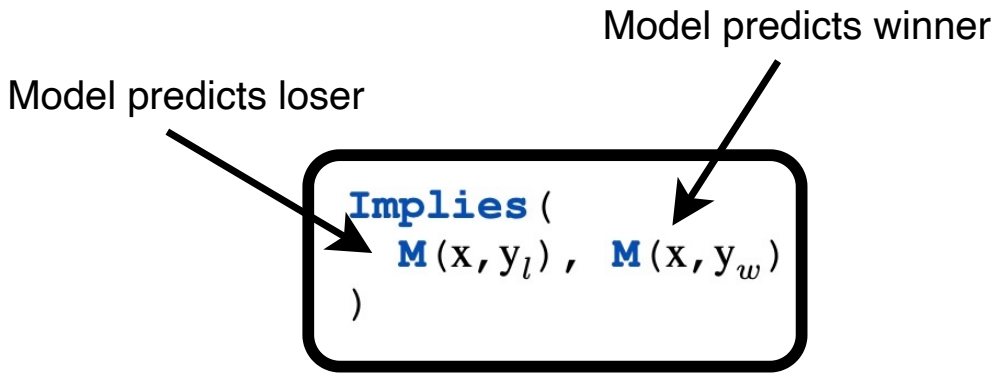
What do these programs tell us?



$\mathrm{M}(x, y) :=$ Model M on input x predicts y

Conceptually: Model predictions are logical propositions, Boolean variables inside of formulae.

What do these programs tell us?

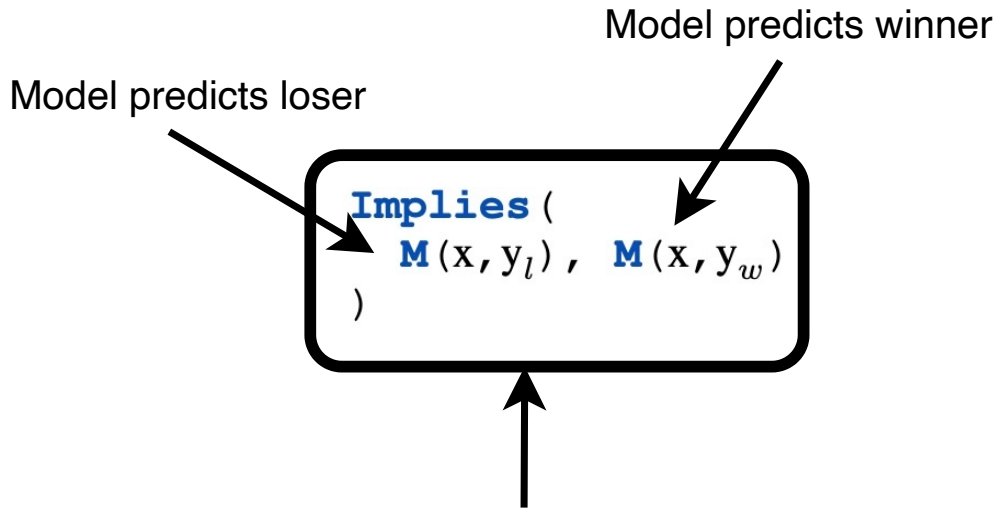


$\mathrm{M}(x, y) :=$ Model M on input x predicts y

$w(\mathrm{M}(x, y)) = \pi_\theta(y \mid x)$

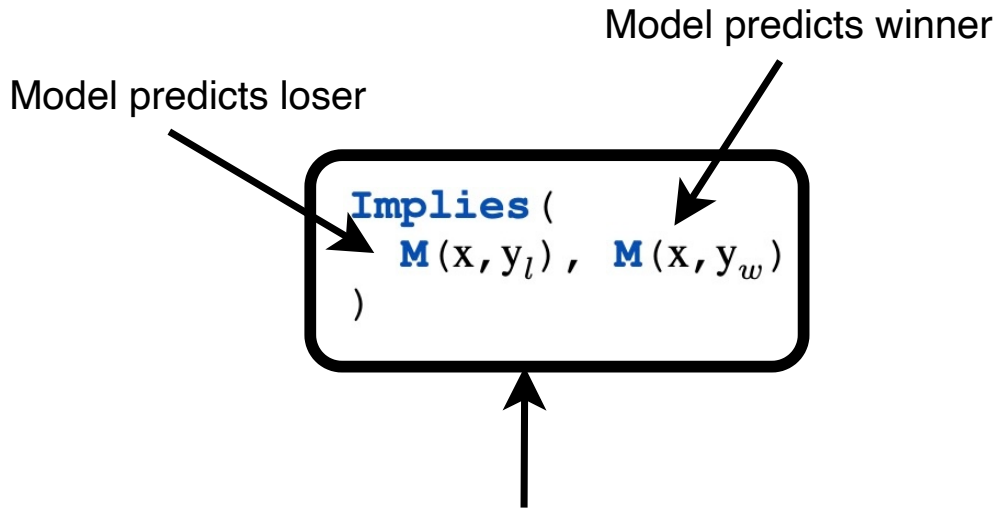
Conceptually: Model predictions are logical propositions, Boolean variables, weighted by prediction probability.

What do these programs tell us?



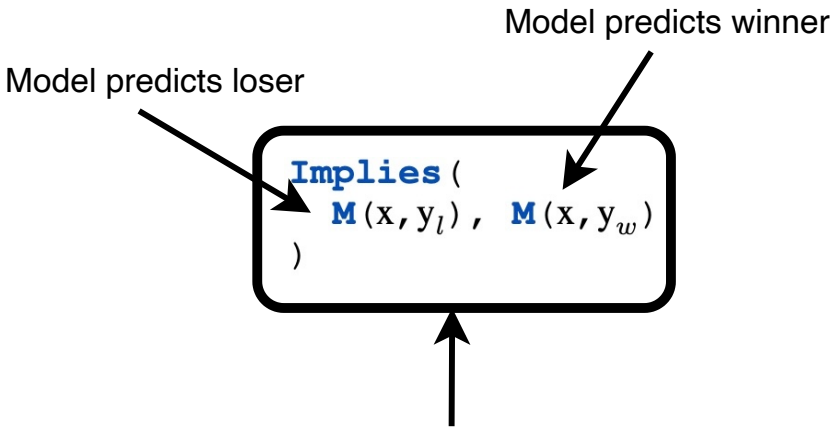
Conceptually: Predictions are connected through Boolean operators, express constraints on predictions; ρ_θ corresponds to a formula.

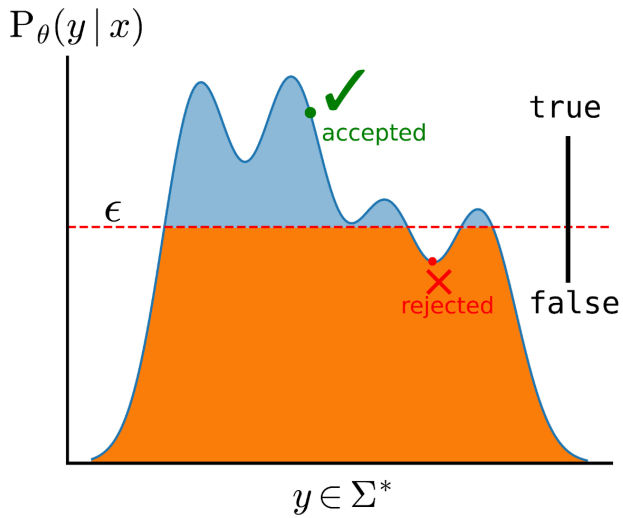
What do these programs tell us?



Assumption: every loss function has an internal logic that can be expressed in this way, we want to uncover that logic.

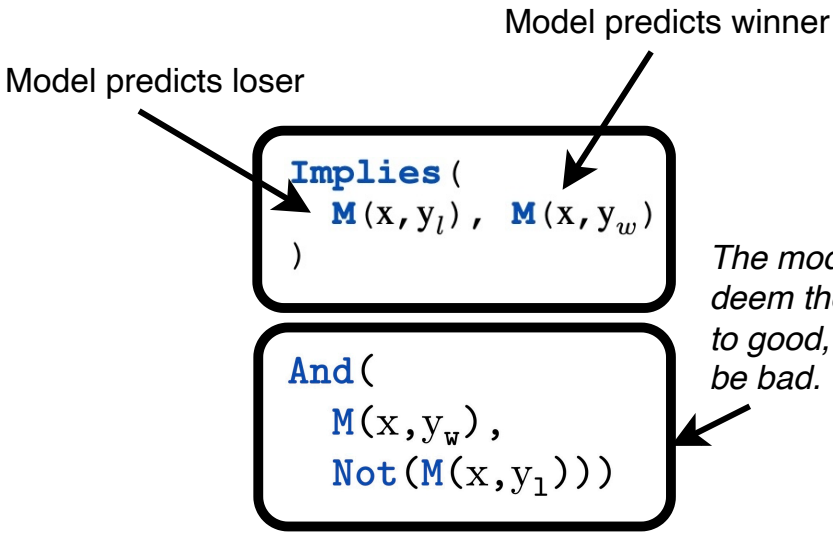
What do these programs tell us?



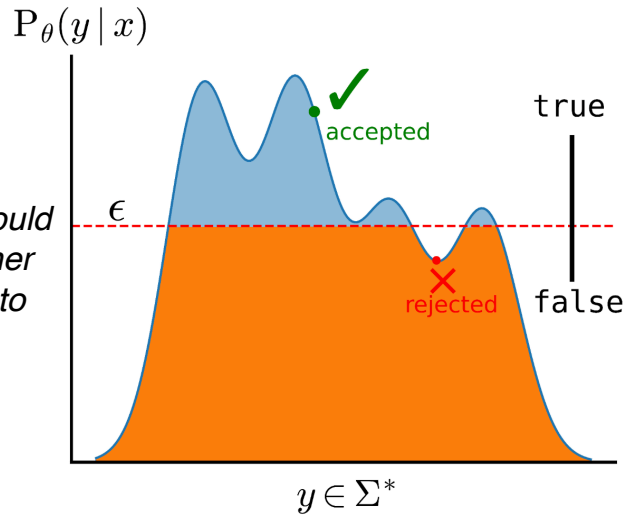


Assumption: every loss function has an internal logic that can be expressed in this way, tells us about the target distribution.

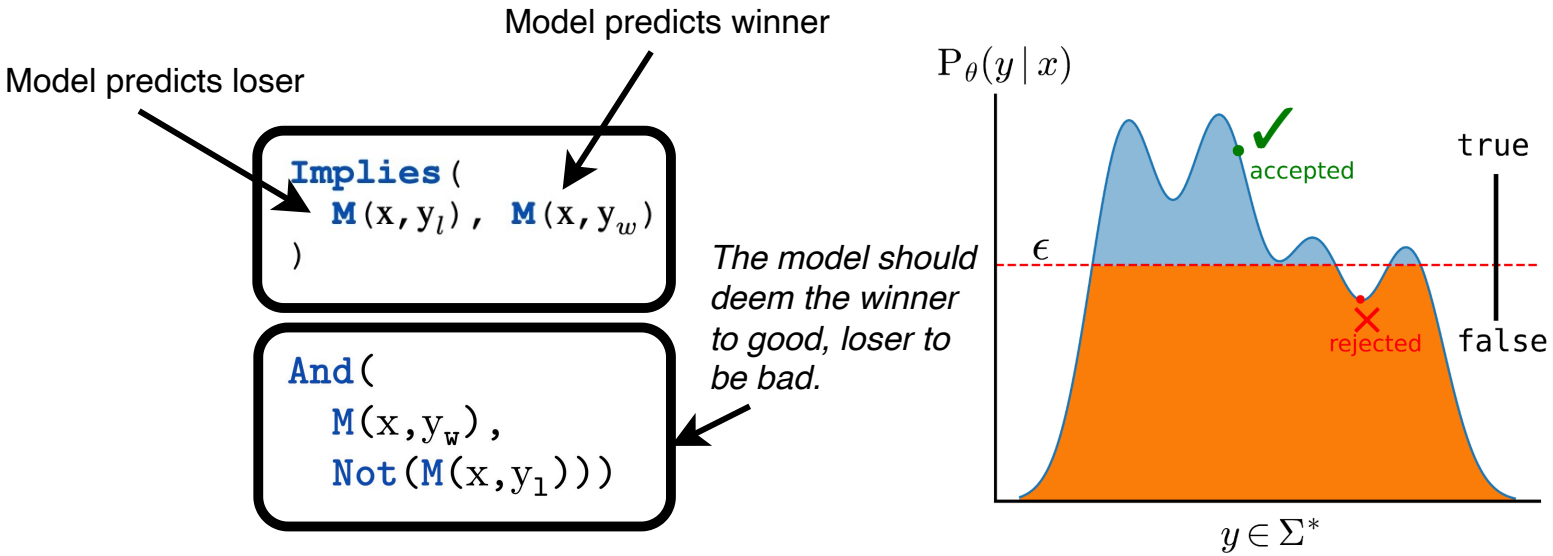
What do these programs tell us?



The model should deem the winner to good, loser to be bad.

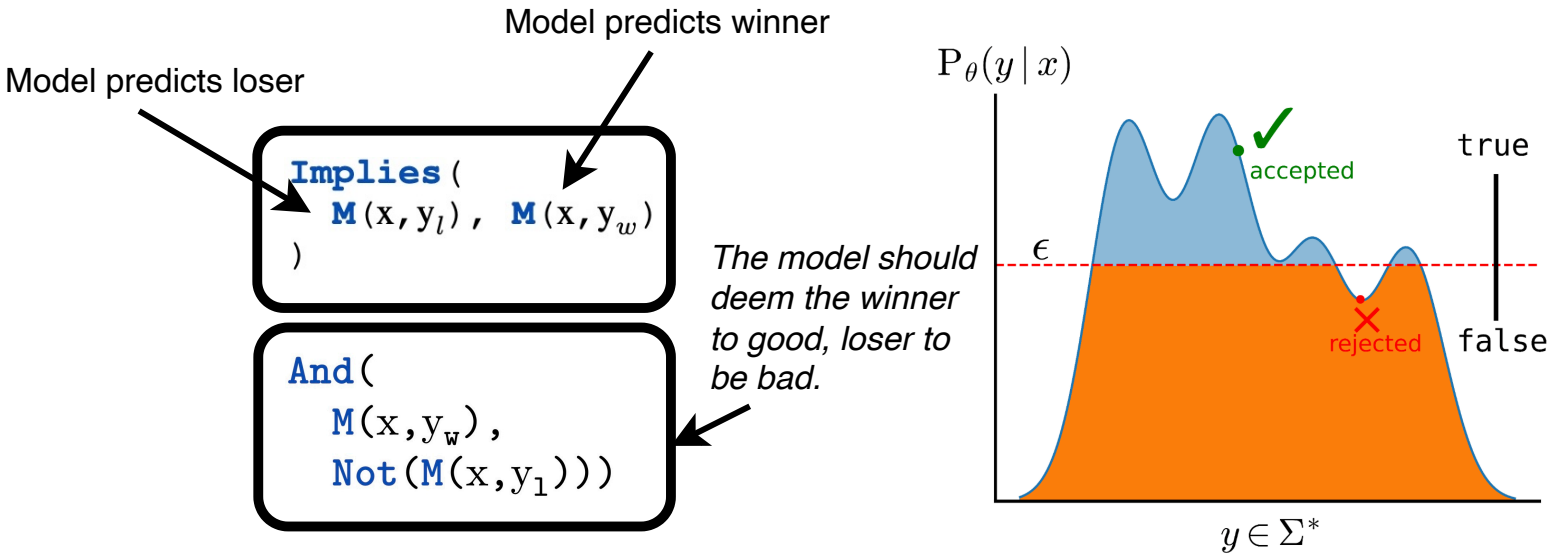


What do these programs tell us?



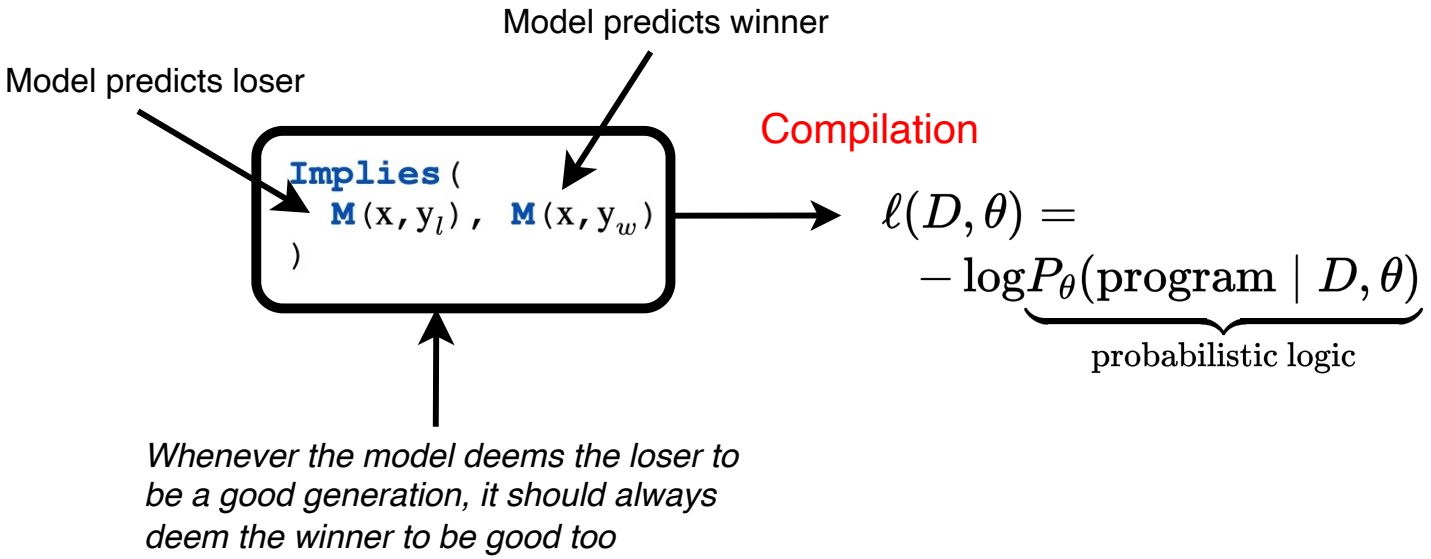
Observation The second program is more strict then the first, can be quantified via semantic entailment.

What do these programs tell us?

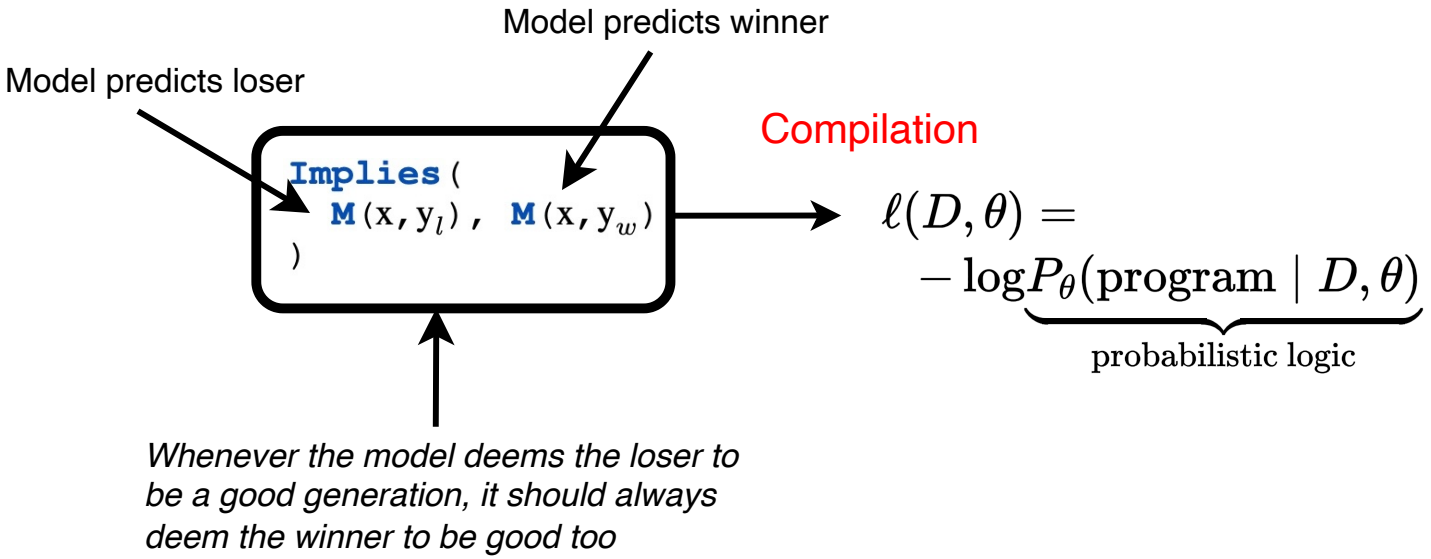


Top program underlies the logic of `SliC` and `CPO` ($s_\theta(x, y_w, y_l)$), bottom is associated with the unlikelihood (Welleck et al., 2019) baseline.

What do these programs tell us?

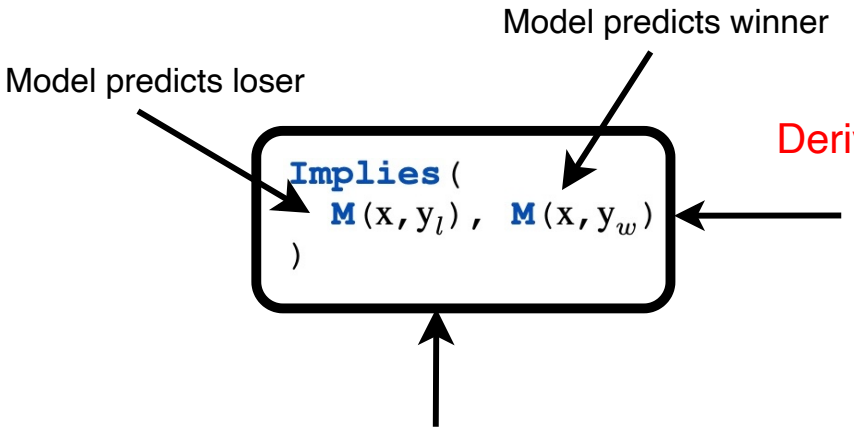


What do these programs tell us?



What we did: Define a novel probabilistic logic for doing this compilation, required some special features.

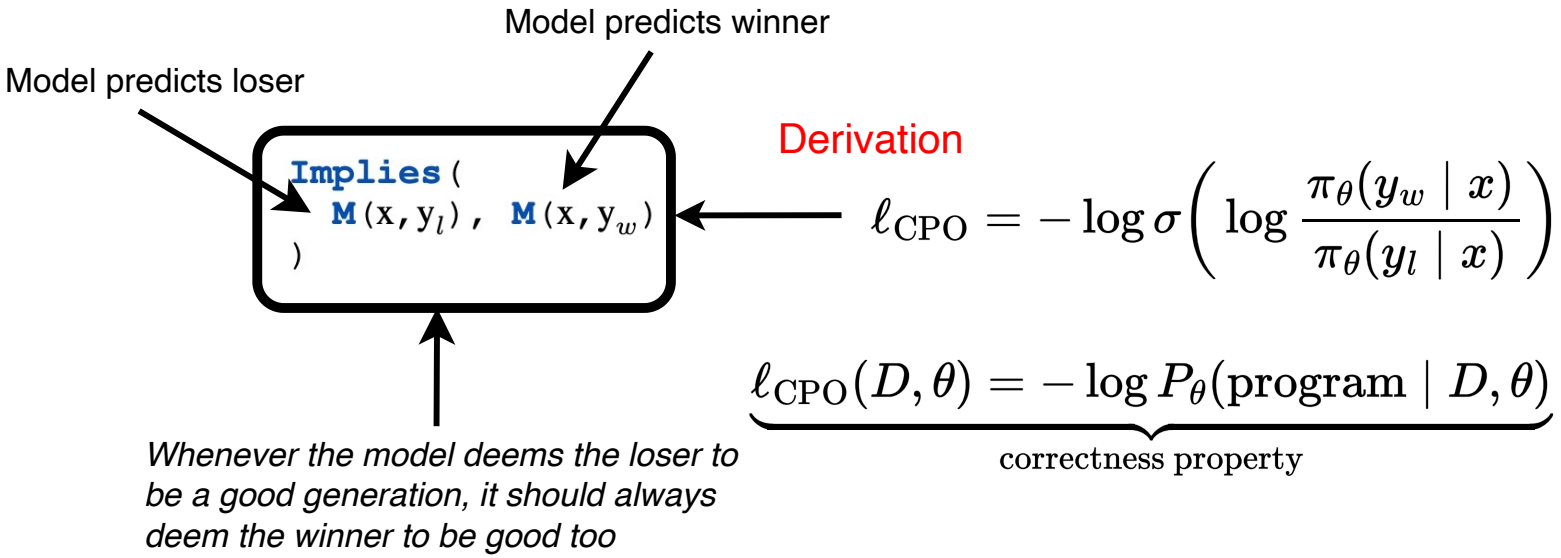
What do these programs tell us?



Derivation

$$\ell_{\text{CPO}} = -\log \sigma\left(\log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$$

What do these programs tell us?

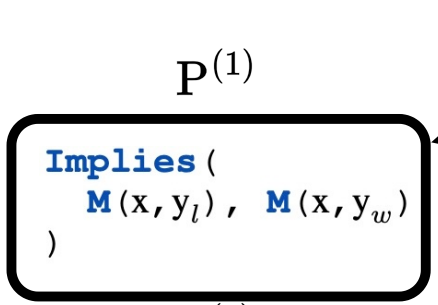


The second thing we did: Defined a mechanical procedure for decompilation, proved its correctness, invariance to choice of f.

How many preference loss functions there?

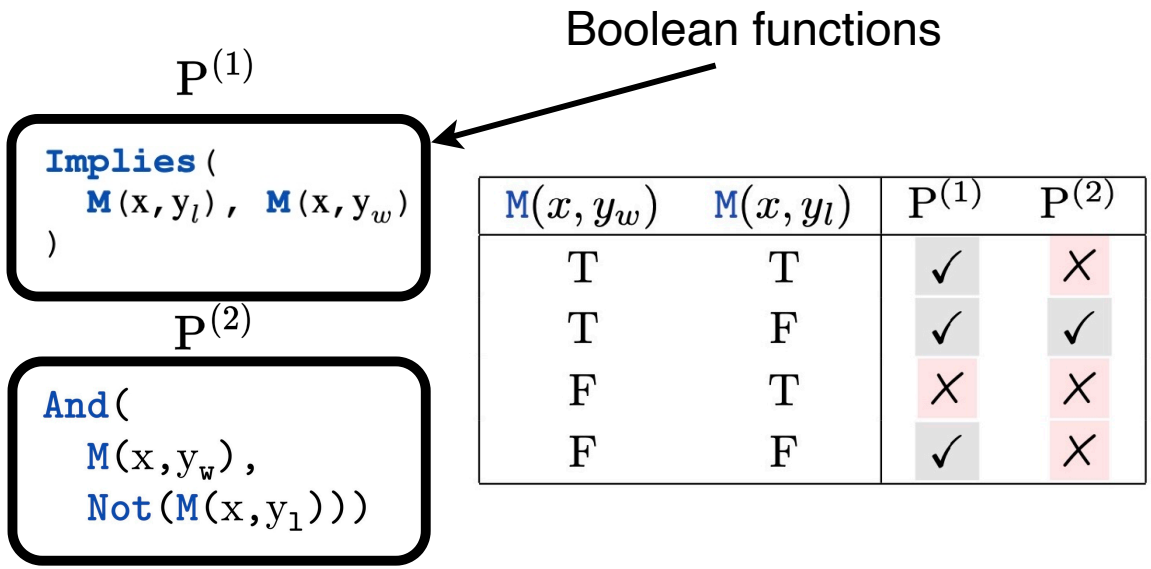
(or How many future DPO papers might be written?)

Why is this useful? understanding the space



$M(x, y_w)$	$M(x, y_l)$	$P^{(1)}$	$P^{(2)}$
T	T	✓	✗
T	F	✓	✓
F	T	✗	✗
F	F	✓	✗

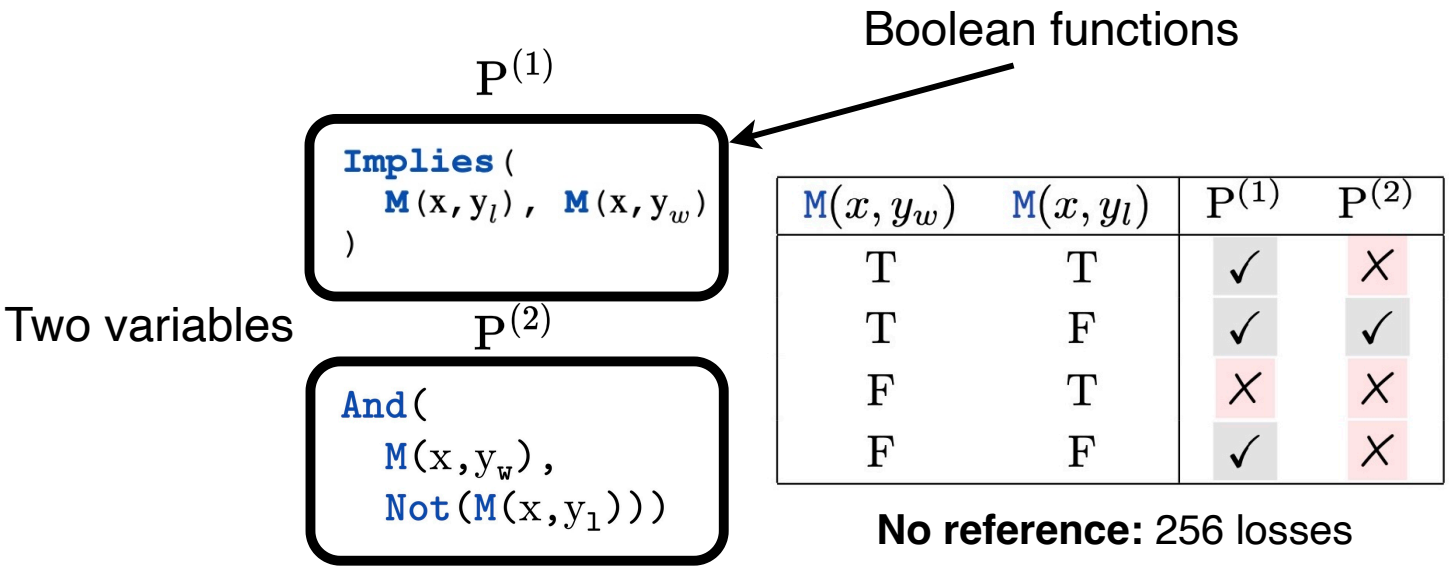
Why is this useful? understanding the space



$\mathrm{M}(x, y_w)$	$\mathrm{M}(x, y_l)$	$\mathrm{P}^{(1)}$	$\mathrm{P}^{(2)}$
T	T	✓	✗
T	F	✓	✓
F	T	✗	✗
F	F	✓	✗

Every program (in our logic) is a pair of Boolean functions (in n variables), can be compiled to a loss, leads to 4^{2^n} possible loss functions.

Why is this useful? understanding the space



Every program (in our logic) is a pair of Boolean functions (in n variables), can be compiled to a loss, leads to 4^{2^n} possible loss functions.

Why is this useful? understanding the space

```
P_DPO := Implies(
  And(Ref(x, y_w), M(x, y_l)),
  And(Ref(x, y_l), M(x, y_w))
)
```

Four variables

$\texttt{Ref}(x, y_w)$	$\texttt{M}(x, y_l)$	$\texttt{Ref}(x, y_l)$	$\texttt{M}(x, y_w)$
F	F	F	F
F	F	T	F
F	F	T	T
F	T	F	F
F	T	F	T
F	T	T	F
F	T	T	T
T	F	F	F
T	F	F	T
T	F	T	F
T	F	T	T
T	T	F	F
T	T	F	T
T	T	T	F
T	T	T	T

With reference: 4,294,967,296 losses

Every program (in our logic) is a pair of Boolean functions (in n variables), can be compiled to a loss, leads to 4^{2^n} possible loss functions.

Answer: loads.

How are loss functions related to one another?

Why is this useful? understanding its structure

$\mathrm{P}^{(1)}$

```
Implies(
  M(x,y_l), M(x,y_w)
)
```

$\mathrm{P}^{(2)}$

```
And(
  M(x,y_w),
  Not(M(x,y_1)))
```

$\mathtt{M}(x, y_w)$	$\mathtt{M}(x, y_l)$	$\mathrm{P}^{(1)}$	$\mathrm{P}^{(2)}$
T	T	✓	✗
T	F	✓	✓
F	T	✗	✗
F	F	✓	✗

Proposition: Loss behavior is monotonic w.r.t semantic entailment,

formally: If $\mathrm{P}^{(2)} \models \mathrm{P}^{(1)}$, then $\ell(D, \theta, \mathrm{P}_2) \geq \ell(D, \theta, \mathrm{P}_1)$ for any D, θ.

Why is this useful? understanding its structure

$\mathrm{P}^{(1)}$

```
Implies(
  M(x,y_l), M(x,y_w)
)
```

$\mathrm{P}^{(2)}$

```
And(
  M(x,y_w),
  Not(M(x,y_1)))
```

$\mathtt{M}(x, y_w)$	$\mathtt{M}(x, y_l)$	$\mathrm{P}^{(1)}$	$\mathrm{P}^{(2)}$
T	T	✓	✗
T	F	✓	✓
F	T	✗	✗
F	F	✓	✗

Proposition: Loss is equivalent under semantic equivalence, i.e., If $\mathrm{P}^{(2)} \equiv \mathrm{P}^{(1)}$ then $\ell(D, \theta, \mathrm{P}_2) = \ell(D, \theta, \mathrm{P}_1)$ for any D, θ.

Why is this useful? understanding its structure

$\mathrm{P}^{(1)}$

```
Implies(
  M(x,y_l), M(x,y_w)
)
```

$\mathrm{P}^{(2)}$

```
And(
  M(x,y_w),
  Not(M(x,y_l)))
```

$\mathtt{M}(x, y_w)$	$\mathtt{M}(x, y_l)$	$\mathrm{P}^{(1)}$	$\mathrm{P}^{(2)}$
T	T	✓	✗
T	F	✓	✓
F	T	✗	✗
F	F	✓	✗

Theorem: It will always hold that $\ell(D, \theta, \mathrm{P}^{(2)}) > \ell(D, \theta, \mathrm{P}^{(1)})$ for any D or θ. (The first loss will always occur all the loss of the second, plus more)

Why is this useful? understanding its structure

$\mathrm{P}^{(1)}$

```
Implies(
  M(x,y_l), M(x,y_w)
)
```

$\mathrm{P}^{(2)}$

```
And(
  M(x,y_w),
  Not(M(x,y_l)))
```

$\mathtt{M}(x, y_w)$	$\mathtt{M}(x, y_l)$	$\mathrm{P}^{(1)}$	$\mathrm{P}^{(2)}$
T	T	✓	✗
T	F	✓	✓
F	T	✗	✗
F	F	✓	✗

Practical tool: If we have an approach that empirically works well, we can devise variants, modify semantics to be more/less constrained.

Answer: through their logical semantics.

What does this tell us about existing approaches?

Can we find improved versions of DPO?

Formalization results: no reference losses

Loss	Symbolic Program $\overline{\mathsf{P}}$
CE	$\mathsf{P} := \texttt{M}(\mathrm{x}, \mathrm{y}_w),\ \mathsf{P}_\mathbf{H} := \top$
Unl	$\mathsf{P} := \texttt{And}(\texttt{M}(\mathrm{x}, \mathrm{y}_w),\ \texttt{Not}(\texttt{M}(\mathrm{x}, \mathrm{y}_l)))$ $\mathsf{P}_\mathbf{H} := \top$
CPO	$\mathsf{P} := \texttt{Implies}(\texttt{M}(\mathrm{x}, \mathrm{y}_l), \texttt{M}(\mathrm{x}, \mathrm{y}_w))$;; one−true constraint $\mathsf{P}_\mathbf{H} := \texttt{Or}(\texttt{M}(\mathrm{x}, \mathrm{y}_l),\ \texttt{M}(\mathrm{x}, \mathrm{y}_w))$
ORPO	$\mathsf{P} := \texttt{Implies}(\texttt{M}(\mathrm{x}, \mathrm{y}_\mathrm{l}), \texttt{M}(\mathrm{x}, \mathrm{y}_\mathrm{w}))$;; one−hot constraint $\mathsf{P}_\mathbf{H} := \texttt{Or}($ $\texttt{And}(\texttt{M}(\mathrm{x}, \mathrm{y}_l),\ \texttt{Not}(\texttt{M}(\mathrm{x}, \mathrm{y}_w))),$ $\texttt{And}(\texttt{Not}(\texttt{M}(\mathrm{x}, \mathrm{y}_l)),\ \texttt{M}(\mathrm{x}, \mathrm{y}_w)))$

Formalization results: no reference losses

Loss	Symbolic Program $\overline{\mathsf{P}}$
CE	$\mathsf{P} := \texttt{M}(\mathrm{x},\mathrm{y}_w),\ \mathsf{P}_\mathbf{H} := \top$
Unl	$\mathsf{P} := \texttt{And}(\texttt{M}(\mathrm{x},\mathrm{y}_w),\ \texttt{Not}(\texttt{M}(\mathrm{x},\mathrm{y}_l)))$ $\mathsf{P}_\mathbf{H} := \top$
CPO	$\mathsf{P} := \texttt{Implies}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\ \texttt{M}(\mathrm{x},\mathrm{y}_w))$;; one−true constraint $\mathsf{P}_\mathbf{H} := \texttt{Or}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\ \texttt{M}(\mathrm{x},\mathrm{y}_w))$
ORPO	$\mathsf{P} := \texttt{Implies}(\texttt{M}(\mathrm{x},\mathrm{y}_\mathrm{l}),\texttt{M}(\mathrm{x},\mathrm{y}_\mathrm{w}))$;; one−hot constraint $\mathsf{P}_\mathbf{H} := \texttt{Or}($ $\texttt{And}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\ \texttt{Not}(\texttt{M}(\mathrm{x},\mathrm{y}_w))),$ $\texttt{And}(\texttt{Not}(\texttt{M}(\mathrm{x},\mathrm{y}_l)),\ \texttt{M}(\mathrm{x},\mathrm{y}_w)))$

Theorem: these programs are correct formalizations of the corresponding losses under our probabilistic logic.

Formalization results: no reference losses

Loss	Symbolic Program $\overline{\mathsf{P}}$
CE	$\mathsf{P} := \texttt{M}(\mathrm{x},\mathrm{y}_w),\ \mathsf{P}_\mathbf{H} := \top$
Unl	$\mathsf{P} := \texttt{And}(\texttt{M}(\mathrm{x},\mathrm{y}_w),\ \texttt{Not}(\texttt{M}(\mathrm{x},\mathrm{y}_l)))$ $\mathsf{P}_\mathbf{H} := \top$
CPO	$\mathsf{P} := \texttt{Implies}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\ \texttt{M}(\mathrm{x},\mathrm{y}_w))$;; one−true constraint $\mathsf{P}_\mathbf{H} := \texttt{Or}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\ \texttt{M}(\mathrm{x},\mathrm{y}_w))$
ORPO	$\mathsf{P} := \texttt{Implies}(\texttt{M}(\mathrm{x},\mathrm{y}_\mathrm{l}),\texttt{M}(\mathrm{x},\mathrm{y}_\mathrm{w}))$;; one−hot constraint $\mathsf{P}_\mathbf{H} := \texttt{Or}($ $\texttt{And}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\ \texttt{Not}(\texttt{M}(\mathrm{x},\mathrm{y}_w))),$ $\texttt{And}(\texttt{Not}(\texttt{M}(\mathrm{x},\mathrm{y}_l)),\ \texttt{M}(\mathrm{x},\mathrm{y}_w)))$

our logic: loss functions are represented by a propositional formula P coupled with a set of hard constraints.

Formalization results: no reference losses

Loss	Symbolic Program $\overline{\mathsf{P}}$
CE	$\mathsf{P} := \texttt{M}(\mathrm{x},\mathrm{y}_w),\ \mathsf{P_H} := \top$
Unl	$\mathsf{P} := \texttt{And}(\texttt{M}(\mathrm{x},\mathrm{y}_w),\ \texttt{Not}(\texttt{M}(\mathrm{x},\mathrm{y}_l)))$ $\mathsf{P_H} := \top$
CPO	$\mathsf{P} := \texttt{Implies}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\ \texttt{M}(\mathrm{x},\mathrm{y}_w))$;; one−true constraint $\mathsf{P_H} := \texttt{Or}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\ \texttt{M}(\mathrm{x},\mathrm{y}_w))$
ORPO	$\mathsf{P} := \texttt{Implies}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\texttt{M}(\mathrm{x},\mathrm{y}_w))$;; one−hot constraint $\mathsf{P_H} := \texttt{Or}($ $\texttt{And}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\ \texttt{Not}(\texttt{M}(\mathrm{x},\mathrm{y}_w))),$ $\texttt{And}(\texttt{Not}(\texttt{M}(\mathrm{x},\mathrm{y}_l)),\ \texttt{M}(\mathrm{x},\mathrm{y}_w)))$

Whenever the model deems the loser to be good, it should always deem the winner to be good.

- This semantics comes up repeatedly, even when we change the underlying logic used for compilation (e.g., to fuzzy logic).

Formalization results: no reference losses

Loss	Symbolic Program $\overline{\mathsf{P}}$
CE	$\mathsf{P} := \texttt{M}(\mathrm{x},\mathrm{y}_w),\ \mathsf{P_H} := \top$
Unl	$\mathsf{P} := \texttt{And}(\texttt{M}(\mathrm{x},\mathrm{y}_w),\ \texttt{Not}(\texttt{M}(\mathrm{x},\mathrm{y}_l)))$ $\mathsf{P_H} := \top$
CPO	$\mathsf{P} := \texttt{Implies}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\ \texttt{M}(\mathrm{x},\mathrm{y}_w))$;; one−true constraint $\mathsf{P_H} := \texttt{Or}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\ \texttt{M}(\mathrm{x},\mathrm{y}_w))$
ORPO	$\mathsf{P} := \texttt{Implies}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\texttt{M}(\mathrm{x},\mathrm{y}_w))$;; one−hot constraint $\mathsf{P_H} := \texttt{Or}($ $\texttt{And}(\texttt{M}(\mathrm{x},\mathrm{y}_l),\ \texttt{Not}(\texttt{M}(\mathrm{x},\mathrm{y}_w))),$ $\texttt{And}(\texttt{Not}(\texttt{M}(\mathrm{x},\mathrm{y}_l)),\ \texttt{M}(\mathrm{x},\mathrm{y}_w)))$

Either the winner or loser is true, or both

Either the winner or loser is true, but not both

- Losses differ in terms of **hard constraints** they impose, shape the structure of the probability space, peculiar properties.

A bit more about the semantics and compilation...

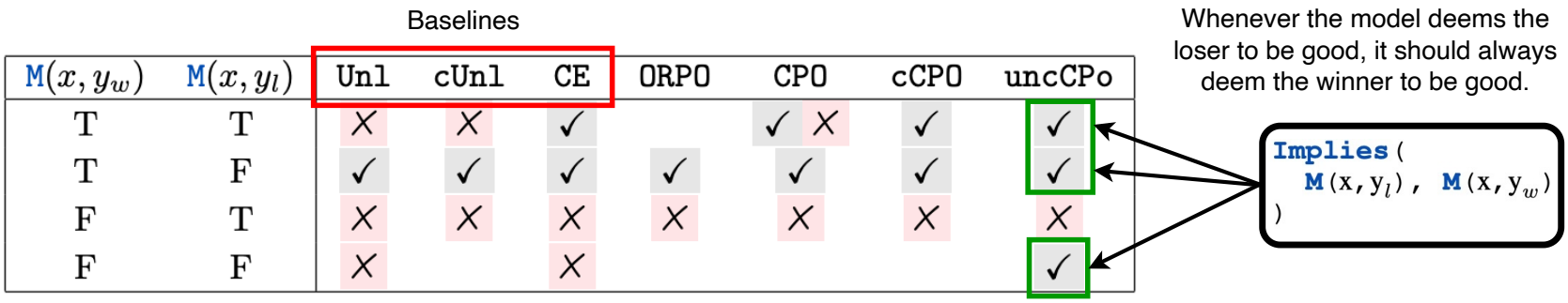
Baselines

M(x, y_w)	M(x, y_l)	Unl	cUnl	CE	ORPO	CPO	cCPO	uncCPo
T	T	✗	✗	✓		✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗
F	F	✗		✗				✓

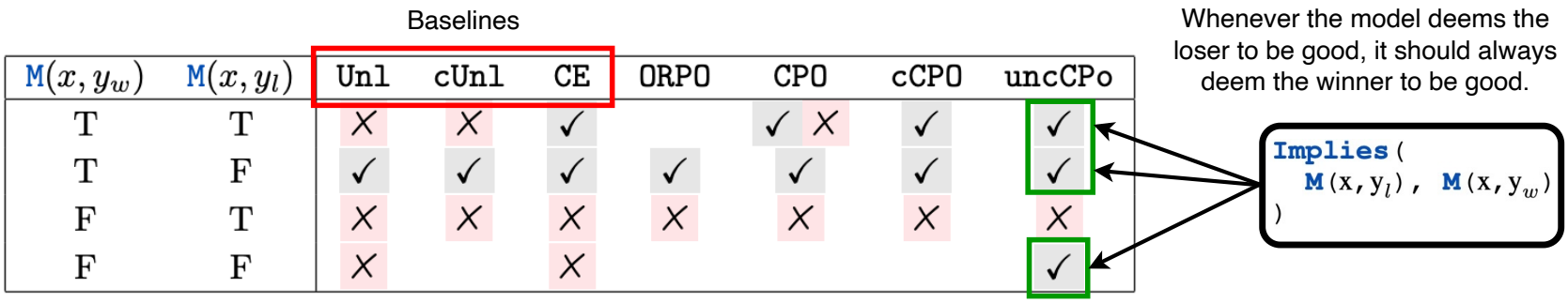
Whenever the model deems the loser to be good, it should always deem the winner to be good.

```
Implies(
  M(x, y_l), M(x, y_w)
)
```

A bit more about the semantics and compilation...



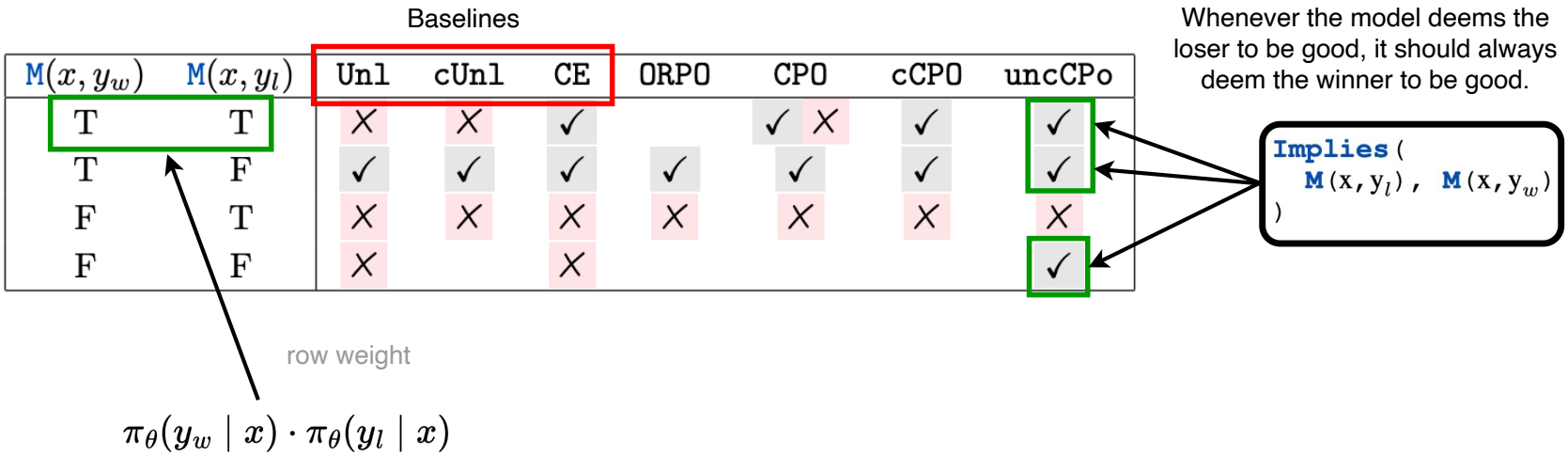
A bit more about the semantics and compilation...



$M(x, y_w)$	$M(x, y_l)$	Unl	cUnl	CE	ORPO	CPO	cCPO	uncCPo
T	T	✗	✗	✓		✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗
F	F	✗		✗				✓

Conventional logical and probabilistic semantics, based on counting the number of rows in a truth table with ✓ .

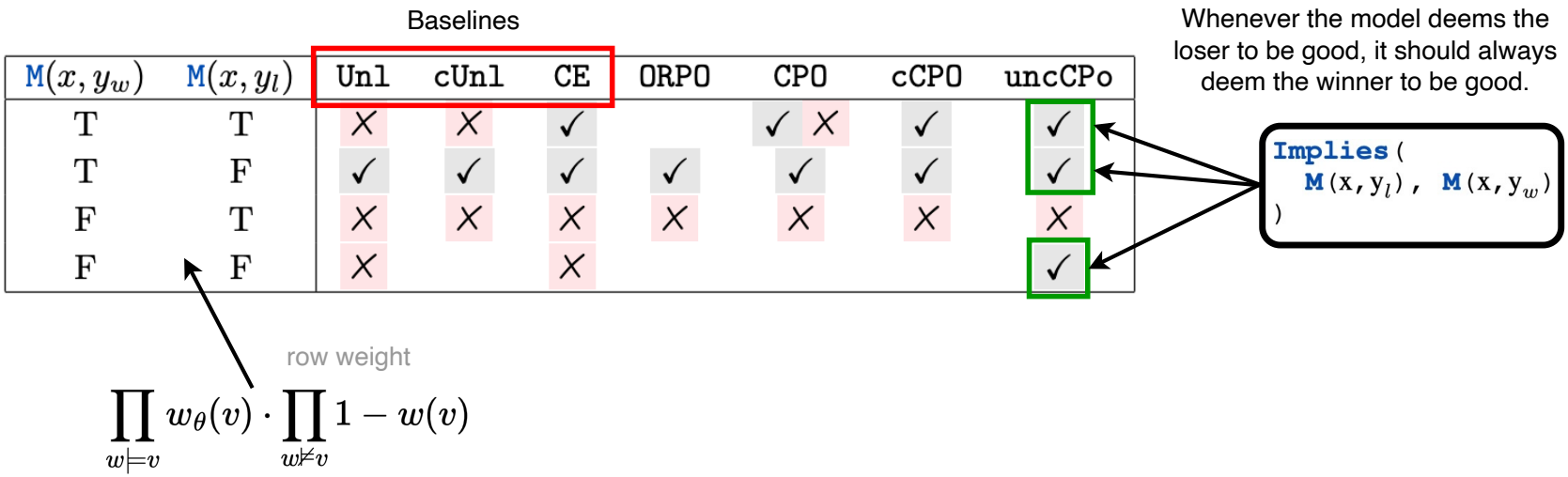
A bit more about the semantics and compilation...



$M(x, y_w)$	$M(x, y_l)$	Unl	cUnl	CE	ORPO	CPO	cCPO	uncCPo
T	T	✗	✗	✓		✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗
F	F	✗		✗				✓

Conventional logical and probabilistic semantics, based on counting the number of rows in a truth table with ✓.

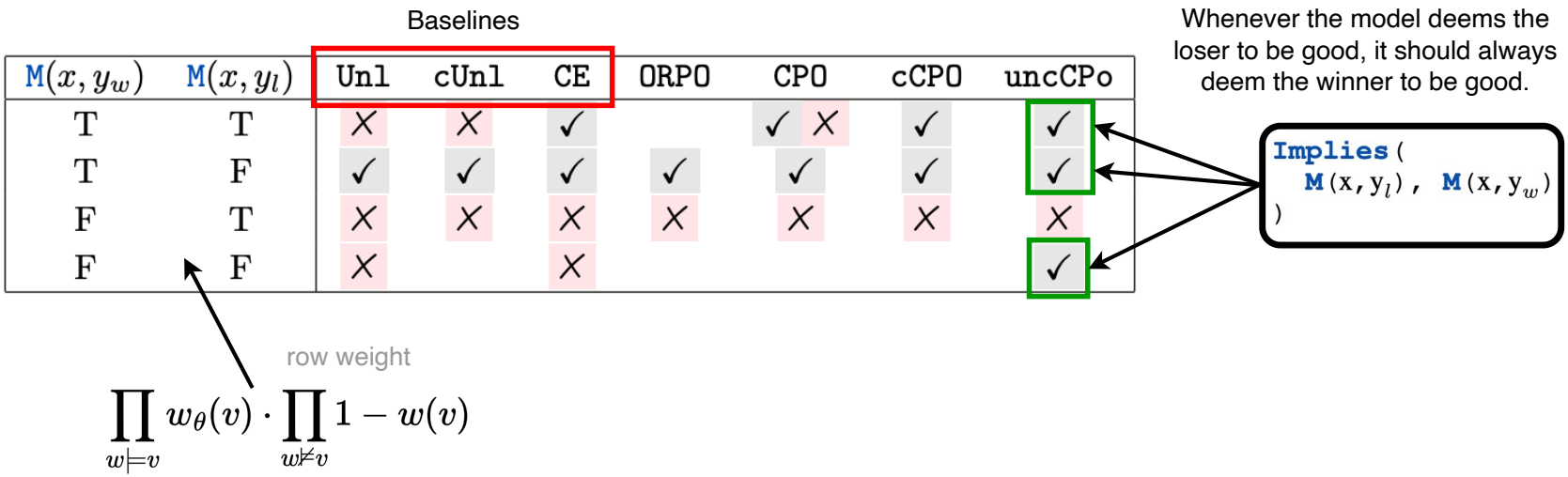
A bit more about the semantics and compilation...



$M(x, y_w)$	$M(x, y_l)$	Unl	cUnl	CE	ORPO	CPO	cCPO	uncCPo
T	T	✗	✗	✓		✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗
F	F	✗		✗				✓

Conventional logical and probabilistic semantics, based on counting the number of rows in a truth table with ✓ .

A bit more about the semantics and compilation...

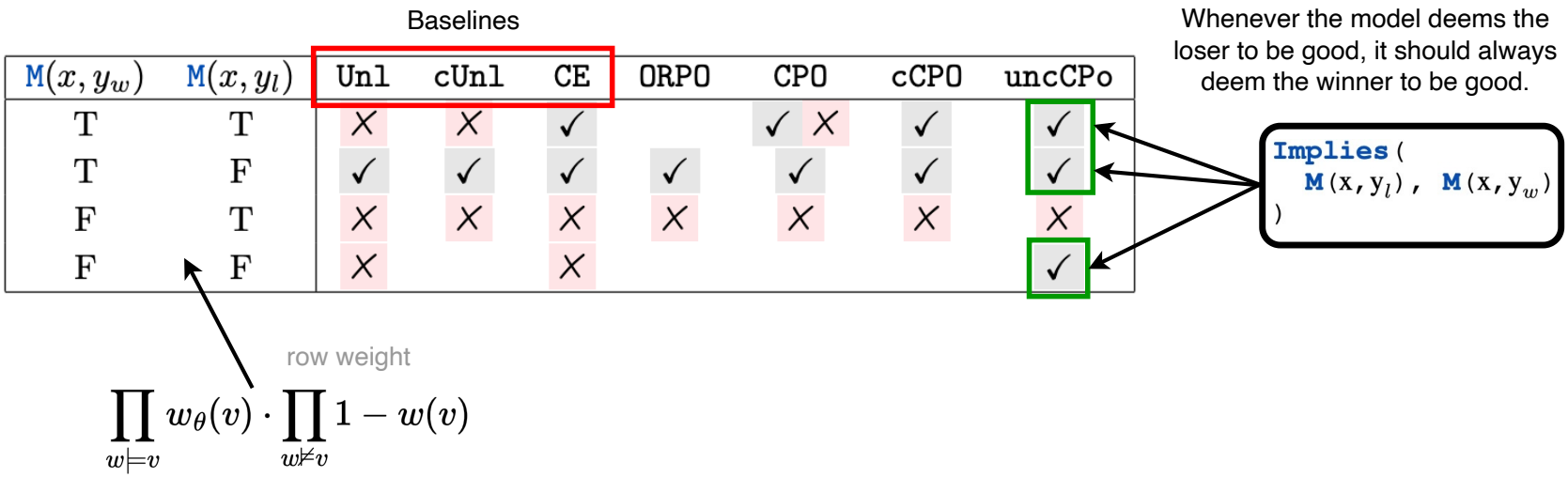


$\texttt{M}(x, y_w)$	$\texttt{M}(x, y_l)$	Unl	cUnl	CE	ORPO	CPO	cCPO	uncCPo
T	T	✗	✗	✓		✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗
F	F	✗		✗				✓

Weighted model counting: how we get a final loss ℓ

$$\ell = -\log P_\theta(\text{program}) = -\log \sigma\left(\log \frac{\text{WCOUNT}(\ \checkmark\ )}{\text{WCOUNT}(\ \times\ )}\right)$$

A bit more about the semantics and compilation...



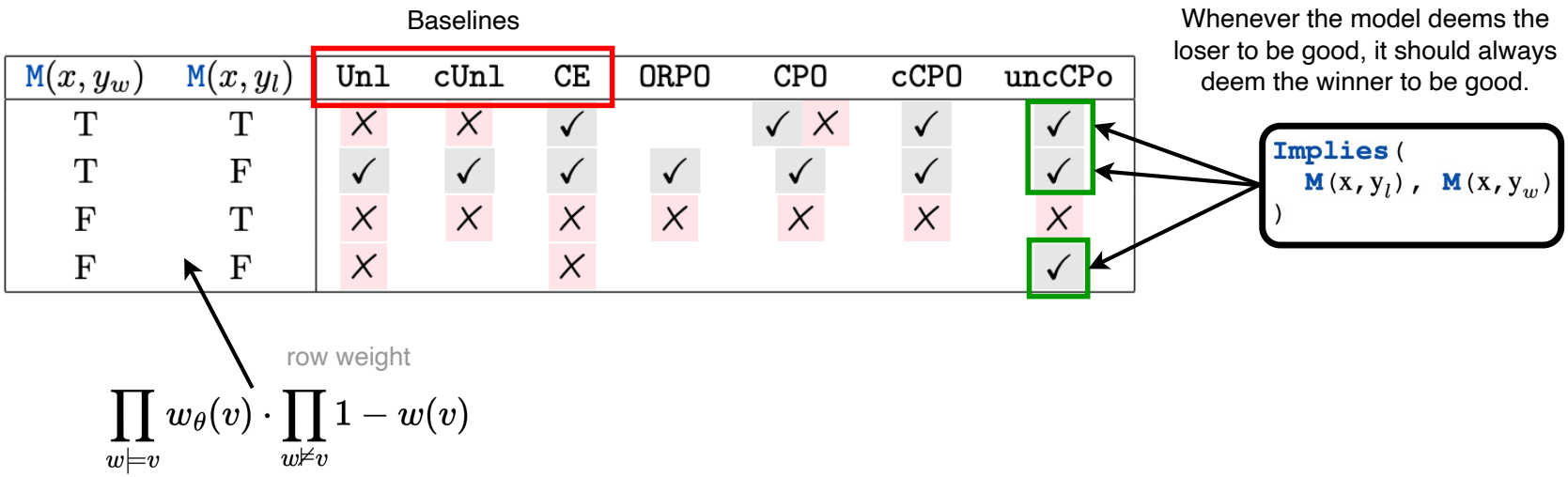
$\texttt{M}(x, y_w)$	$\texttt{M}(x, y_l)$	Unl	cUnl	CE	ORPO	CPO	cCPO	uncCPo
T	T	✗	✗	✓		✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗
F	F	✗		✗				✓

Weighted model counting: how we get a final loss ℓ

$$\ell = -\log P_\theta(\text{program}) = -\log \sigma\left(\log \frac{\text{WCOUNT}(\ \checkmark\ )}{\text{WCOUNT}(\ \times\ )}\right)$$

Well studied problem, basis for much of classical probabilistic inference, PGMs (Chavira and Darwiche, 2008), semantic loss (Xu et al., 2018).

A bit more about the semantics and compilation...



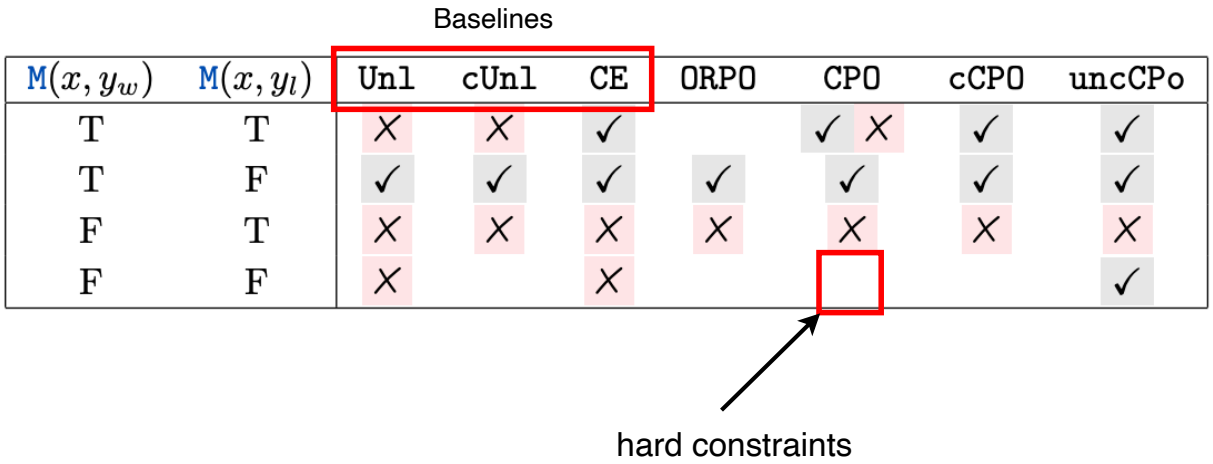
$M(x, y_w)$	$M(x, y_l)$	Unl	cUnl	CE	ORPO	CPO	cCPO	uncCPo
T	T	✗	✗	✓		✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗
F	F	✗		✗				✓

Weighted model counting: how we get a final loss ℓ

$$\ell = -\log P_\theta(\text{program}) = -\log \sigma\left(\log \frac{\text{WCOUNT}(\ \checkmark\ )}{\text{WCOUNT}(\ \times\ )}\right)$$

Side note: close to semantics in preference logic, useful for structured probabilistic inference in LLMs (Kassner et al., 2023; Gu et al., 2023).

A bit more about the semantics and compilation...



Whenever the model deems the loser to be good, it should always deem the winner to be good.

```
Implies(
  M(x, y_l), M(x, y_w)
)
```

Weighted model counting: how we get a final loss ℓ

$$\ell = -\log P_\theta(\text{program}) = -\log \sigma\left(\log \frac{\text{WCOUNT}(\ \checkmark\ )}{\text{WCOUNT}(\ \times\ )}\right)$$

Important extra details: Ability to add hard constraints, arbitrary normalization terms (related to pair of Boolean functions).

A bit more about the semantics and compilation...

Baselines

M(x, y_w)	M(x, y_l)	Unl	cUnl	CE	ORPO	CPO	cCPO	uncCPo
T	T	✗	✗	✓		✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗
F	F	✗		✗				✓

non-standard normalization

Whenever the model deems the loser to be good, it should always deem the winner to be good.

```
Implies(
  M(x, y_l), M(x, y_w)
)
```

Weighted model counting: how we get a final loss ℓ

$$\ell = -\log P_\theta(\text{program}) = -\log \sigma\left(\log \frac{\text{WCOUNT}(\ \checkmark\ )}{\text{WCOUNT}(\ \times\ )}\right)$$

Important extra details: Ability to add hard constraints, arbitrary normalization terms (related to pair of Boolean functions).

A bit more about the semantics and compilation...

Baselines

$\texttt{M}(x, y_w)$	$\texttt{M}(x, y_l)$	Unl	cUnl	CE	ORPO	CPO	cCPO	uncCPo
T	T	✗	✗	✓		✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗
F	F	✗		✗				✓

Whenever the model deems the loser to be good, it should always deem the winner to be good.

```
Implies(
  M(x, y_l), M(x, y_w)
)
```

Weighted model counting: how we get a final loss ℓ

$$-\log P_\theta(\text{program}_{\text{CPO}}) = -\log \sigma\left(\log \frac{\text{WCOUNT}(\ \checkmark\ )}{\text{WCOUNT}(\ \times\ )}\right)$$

$$= \underbrace{-\log \sigma\left(\log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)}_{\ell_{\text{CPO}}}$$

Looks can be misleading...

Baselines

$\texttt{M}(x, y_w)$	$\texttt{M}(x, y_l)$	Unl	cUnl	CE	ORPO	CPO	cCPO	uncCPo
T	T	✗	✗	✓		✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗
F	F	✗		✗				✓

novel loss

Whenever the model deems the loser to be good, it should always deem the winner to be good.

```
Implies(
  M(x, y_l), M(x, y_w)
)
```

Model model counting

$$-\log P_\theta(\text{program}_{\text{uncCPo}}) = -\log \sigma\left(\log \frac{\text{WCOUNT}(\ \checkmark\ )}{\text{WCOUNT}(\ \times\ )}\right)$$

$$= \underbrace{-\log \sigma\left(\log \frac{\pi_\theta(y_l \mid x)\pi_\theta(y_w \mid x) + (1 - \pi_\theta(y_l \mid x))}{\pi_\theta(y_l \mid x)(1 - \pi_\theta(y_w \mid x))}\right)}_{\ell_{\text{unCPo}}}$$

Looks can be misleading...

Baselines

$\texttt{M}(x, y_w)$	$\texttt{M}(x, y_l)$	Unl	cUnl	CE	ORPO	CPO	cCPO	uncCPo
T	T	✗	✗	✓		✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗
F	F	✗		✗				✓

novel loss

Whenever the model deems the loser to be good, it should always deem the winner to be good.

```
Implies(
  M(x, y_l), M(x, y_w)
)
```

Model model counting

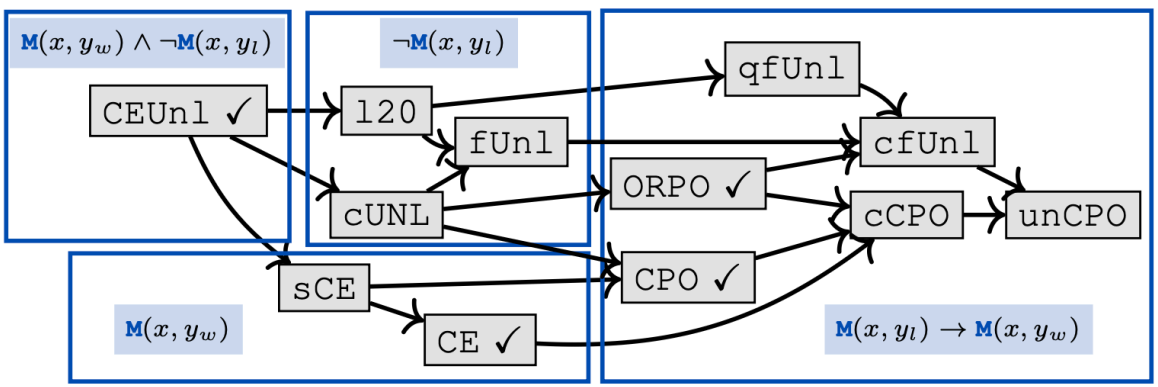
$$-\log P_\theta(\text{program}_{\text{uncCPo}}) = -\log \sigma\left(\log \frac{\text{WCOUNT}(\ \checkmark\ )}{\text{WCOUNT}(\ \times\ )}\right)$$

$$= \underbrace{-\log \sigma\left(\log \frac{\pi_\theta(y_l \mid x)\pi_\theta(y_w \mid x) + (1 - \pi_\theta(y_l \mid x))}{\pi_\theta(y_l \mid x)(1 - \pi_\theta(y_w \mid x))}\right)}_{\ell_{\text{unCPo}}}$$

This is entirely justifiable semantically, but probably impossible to derive.

The no reference loss landscape

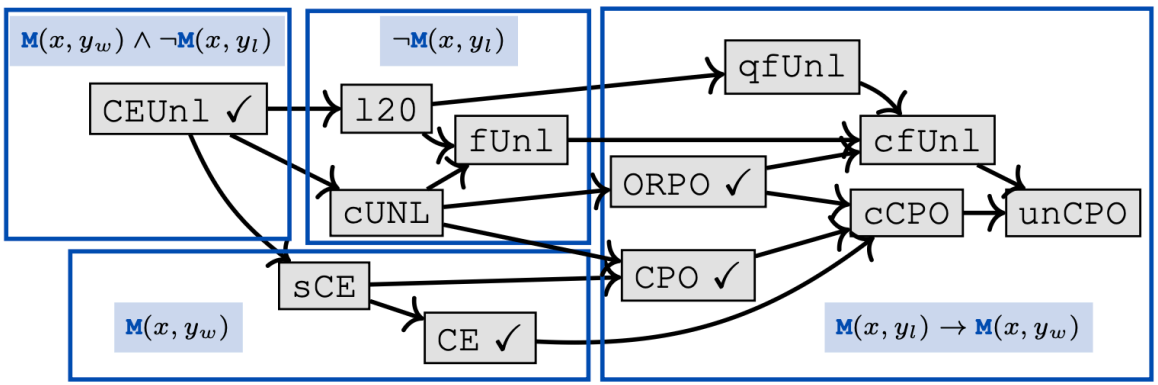
$\mathtt{M}(x, y_w)$	$\mathtt{M}(x, y_l)$	Unl	qfUnl	cUnl	CE	bCE	fUnl	ORPO	cfUnl	CPO	cCPO	uncCPo
T	T	✗		✗	✓	✓	✗			✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
F	F	✗	✓ ✗		✗	✓ ✗	✓		✓			✓



- **Loss entailment graph**: Each edge indicates a strict semantic entailment relation between two losses, more constrained as you move right.

The no reference loss landscape

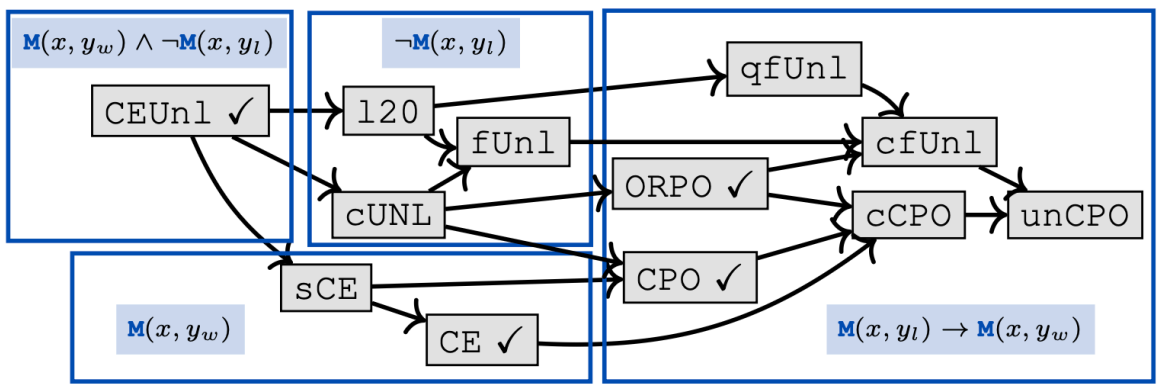
$\mathtt{M}(x, y_w)$	$\mathtt{M}(x, y_l)$	Unl	qfUnl	cUnl	CE	bCE	fUnl	ORPO	cfUnl	CPO	cCPO	uncCPo
T	T	✗		✗	✓	✓	✗			✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
F	F	✗	✓ ✗		✗	✓ ✗	✓		✓			✓



Theorem: For any D and θ, it will hold that $\ell_{\mathtt{CEUnl}}(D, \theta) > \ell_{\mathtt{CE}}(D, \theta)$ and $\ell_{\mathtt{Unl}}(D, \theta) > \ell_{\mathtt{ORPO}}(D, \theta)$ and $\ell_{\mathtt{Unl}}(D, \theta) > \ell_{\mathtt{CPO}}(D, \theta)$.

The no reference loss landscape

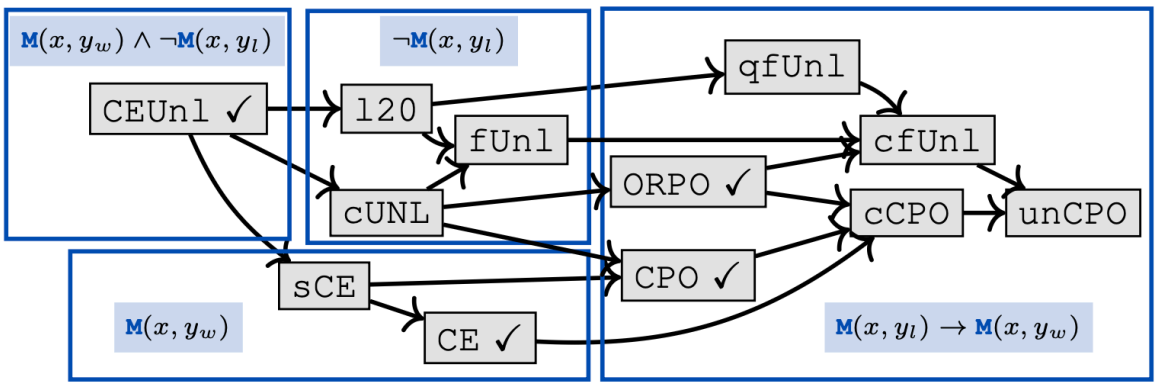
$\texttt{M}(x, y_w)$	$\texttt{M}(x, y_l)$	Unl	qfUnl	cUnl	CE	bCE	fUnl	ORPO	cfUnl	CPO	cCPO	uncCPo
T	T	✗		✗	✓	✓	✗			✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
F	F	✗	✓ ✗		✗	✓ ✗	✓		✓			✓



Observation: There are many novel losses here that haven't been tested, both as variations of baselines and ORPO and CPO, not exhaustive

The no reference loss landscape

$\texttt{M}(x, y_w)$	$\texttt{M}(x, y_l)$	Unl	qfUnl	cUnl	CE	bCE	fUnl	ORPO	cfUnl	CPO	cCPO	uncCPo
T	T	✗		✗	✓	✓	✗			✓ ✗	✓	✓
T	F	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
F	T	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
F	F	✗	✓ ✗		✗	✓ ✗	✓		✓			✓



Questions: What are the trends we see empirically when we work through this graph; is there something special about this middle region?

Are any of these new losses good?

Losses with a reference model

SimPO	P := Implies(And(Mref(x,y_w),M(x,y_l)), And(Mref(x,y_l),M(x,y_w))) P_H := Or(And(Mref(x,y_l),M(x,y_w)), And(Mref(x,y_w),M(x,y_l)))
DPO	P := Implies(And(Ref(x,y_w),M(x,y_l)), And(Ref(x,y_l),M(x,y_w))) P_H := Or(And(Ref(x,y_l),M(x,y_w)), And(Ref(x,y_w),M(x,y_l)))
DPOP	P := Implies(And(Ref(x,y),Ref_1(x,y_w),M(x,y_l)), And(Ref(x,y_l),M(x,y_w), M_1(x,y_w))) P_H := Or(And(Ref(x,y),Ref_1(x,y_w),M(x,y_l)), And(Ref(x,y_l),M(x,y_w), M_1(x,y_w)))

If the reference model deems the winner to be good and our model says that the loser is good, then our model should deem the winner to be good and the reference model should deem the loser is good.

Losses with a reference model

SimPO	$P := \texttt{Implies}(\texttt{And}(\texttt{Mref}(x,y_w),\texttt{M}(x,y_l)),\ \texttt{And}(\texttt{Mref}(x,y_l),\texttt{M}(x,y_w)))$ $P_H := \texttt{Or}(\texttt{And}(\texttt{Mref}(x,y_l),\texttt{M}(x,y_w)),\ \texttt{And}(\texttt{Mref}(x,y_w),\texttt{M}(x,y_l)))$
DPO	$P := \texttt{Implies}(\texttt{And}(\texttt{Ref}(x,y_w),\texttt{M}(x,y_l)),\ \texttt{And}(\texttt{Ref}(x,y_l),\texttt{M}(x,y_w)))$ $P_H := \texttt{Or}(\texttt{And}(\texttt{Ref}(x,y_l),\texttt{M}(x,y_w)),\ \texttt{And}(\texttt{Ref}(x,y_w),\texttt{M}(x,y_l)))$
DPOP	$P := \texttt{Implies}(\texttt{And}(\texttt{Ref}(x,y),\texttt{Ref}_1(x,y_w),\texttt{M}(x,y_l)),\ \texttt{And}(\texttt{Ref}(x,y_l),\texttt{M}(x,y_w),\ \texttt{M}_1(x,y_w)))$ $P_H := \texttt{Or}(\texttt{And}(\texttt{Ref}(x,y),\texttt{Ref}_1(x,y_w),\texttt{M}(x,y_l)),\ \texttt{And}(\texttt{Ref}(x,y_l),\texttt{M}(x,y_w),\ \texttt{M}_1(x,y_w)))$

If the reference model deems the winner to be good and our model says that the loser is good, then our model should deem the winner to be good and the reference model should deem the loser is good.

- **Observation**: Log-ratio difference has a strange semantics, hard to justify, problematic for all variants of DPO we consider.

Losses with a reference model

SimPO	$P := \texttt{Implies}(\texttt{And}(\texttt{Mref}(x,y_w),\texttt{M}(x,y_l)),\ \texttt{And}(\texttt{Mref}(x,y_l),\texttt{M}(x,y_w)))$ $P_H := \texttt{Or}(\texttt{And}(\texttt{Mref}(x,y_l),\texttt{M}(x,y_w)),\ \texttt{And}(\texttt{Mref}(x,y_w),\texttt{M}(x,y_l)))$
DPO	$P := \texttt{Implies}(\texttt{And}(\texttt{Ref}(x,y_w),\texttt{M}(x,y_l)),\ \texttt{And}(\texttt{Ref}(x,y_l),\texttt{M}(x,y_w)))$ $P_H := \texttt{Or}(\texttt{And}(\texttt{Ref}(x,y_l),\texttt{M}(x,y_w)),\ \texttt{And}(\texttt{Ref}(x,y_w),\texttt{M}(x,y_l)))$
DPOP	$P := \texttt{Implies}(\texttt{And}(\texttt{Ref}(x,y),\texttt{Ref}_1(x,y_w),\texttt{M}(x,y_l)),\ \texttt{And}(\texttt{Ref}(x,y_l),\texttt{M}(x,y_w),\ \texttt{M}_1(x,y_w)))$ $P_H := \texttt{Or}(\texttt{And}(\texttt{Ref}(x,y),\texttt{Ref}_1(x,y_w),\texttt{M}(x,y_l)),\ \texttt{And}(\texttt{Ref}(x,y_l),\texttt{M}(x,y_w),\ \texttt{M}_1(x,y_w)))$

If the reference model deems the winner to be good and our model says that the loser is good, then our model should deem the winner to be good and the reference model should deem the loser is good.

same semantics, multiplies magnitude of winner prediction

- **Observation**: Log-ratio difference has a strange semantics, hard to justify, problematic for all variants of DPO we consider.

Losses with a reference model

SimPO	$\mathsf{P} := \mathtt{Implies}(\mathtt{And}(\mathtt{Mref}(\mathrm{x},\mathrm{y}_w),\mathtt{M}(\mathrm{x},\mathrm{y}_l)),\ \mathtt{And}(\mathtt{Mref}(\mathrm{x},\mathrm{y}_l),\mathtt{M}(\mathrm{x},\mathrm{y}_w)))$ $\mathsf{P_H} := \mathtt{Or}(\mathtt{And}(\mathtt{Mref}(\mathrm{x},\mathrm{y}_l),\mathtt{M}(\mathrm{x},\mathrm{y}_w)),\ \mathtt{And}(\mathtt{Mref}(\mathrm{x},\mathrm{y}_w),\mathtt{M}(\mathrm{x},\mathrm{y}_l)))$
DPO	$\mathsf{P} := \mathtt{Implies}(\mathtt{And}(\mathtt{Ref}(\mathrm{x},\mathrm{y}_w),\mathtt{M}(\mathrm{x},\mathrm{y}_l)),\ \mathtt{And}(\mathtt{Ref}(\mathrm{x},\mathrm{y}_l),\mathtt{M}(\mathrm{x},\mathrm{y}_w)))$ $\mathsf{P_H} := \mathtt{Or}(\mathtt{And}(\mathtt{Ref}(\mathrm{x},\mathrm{y}_l),\mathtt{M}(\mathrm{x},\mathrm{y}_w)),\ \mathtt{And}(\mathtt{Ref}(\mathrm{x},\mathrm{y}_w),\mathtt{M}(\mathrm{x},\mathrm{y}_l)))$
DPOP	$\mathsf{P} := \mathtt{Implies}(\mathtt{And}(\mathtt{Ref}(\mathrm{x},\mathrm{y}),\mathtt{Ref}_1(\mathrm{x},\mathrm{y}_\mathrm{w}),\mathtt{M}(\mathrm{x},\mathrm{y}_\mathrm{l})),\ \mathtt{And}(\mathtt{Ref}(\mathrm{x},\mathrm{y}_\mathrm{l}),\mathtt{M}(\mathrm{x},\mathrm{y}_\mathrm{w}),\ \mathtt{M}_1(\mathrm{x},\mathrm{y}_\mathrm{w})))$ $\mathsf{P_H} := \mathtt{Or}(\mathtt{And}(\mathtt{Ref}(\mathrm{x},\mathrm{y}),\mathtt{Ref}_1(\mathrm{x},\mathrm{y}_\mathrm{w}),\mathtt{M}(\mathrm{x},\mathrm{y}_\mathrm{l})),\ \mathtt{And}(\mathtt{Ref}(\mathrm{x},\mathrm{y}_\mathrm{l}),\mathtt{M}(\mathrm{x},\mathrm{y}_\mathrm{w}),\ \mathtt{M}_1(\mathrm{x},\mathrm{y}_\mathrm{w})))$

Either the reference model deems the loser to be good and the main model deems the winner to be good or the reference model deems the winner to be good and the main model deems the loser to be good, or both

- **Observation**: Hard constraints are also very strange, make it not possible to connect formal analysis with no reference losses.

Losses with a reference model

SimPO	$\mathsf{P} := \texttt{Implies}(\texttt{And}(\texttt{Mref}(\mathrm{x},\mathrm{y}_w),\texttt{M}(\mathrm{x},\mathrm{y}_l)),\ \texttt{And}(\texttt{Mref}(\mathrm{x},\mathrm{y}_l),\texttt{M}(\mathrm{x},\mathrm{y}_w)))$ $\mathsf{P_H} := \texttt{Or}(\texttt{And}(\texttt{Mref}(\mathrm{x},\mathrm{y}_l),\texttt{M}(\mathrm{x},\mathrm{y}_w)),\ \texttt{And}(\texttt{Mref}(\mathrm{x},\mathrm{y}_w),\texttt{M}(\mathrm{x},\mathrm{y}_l)))$
DPO	$\mathsf{P} := \texttt{Implies}(\texttt{And}(\texttt{Ref}(\mathrm{x},\mathrm{y}_w),\texttt{M}(\mathrm{x},\mathrm{y}_l)),\ \sout{\texttt{And}(\texttt{Ref}(\mathrm{x},\mathrm{y}_l),\texttt{M}(\mathrm{x},\mathrm{y}_w))})$ $\mathsf{P_H} := \texttt{Or}(\sout{\texttt{And}(\texttt{Ref}(\mathrm{x},\mathrm{y}_l),\texttt{M}(\mathrm{x},\mathrm{y}_w))},\ \texttt{And}(\texttt{Ref}(\mathrm{x},\mathrm{y}_w),\texttt{M}(\mathrm{x},\mathrm{y}_l)))$
DPOP	$\mathsf{P} := \texttt{Implies}(\texttt{And}(\texttt{Ref}(\mathrm{x},\mathrm{y}),\texttt{Ref}_1(\mathrm{x},\mathrm{y}_\mathrm{w}),\texttt{M}(\mathrm{x},\mathrm{y}_\mathrm{l})),\ \texttt{And}(\texttt{Ref}(\mathrm{x},\mathrm{y}_\mathrm{l}),\texttt{M}(\mathrm{x},\mathrm{y}_\mathrm{w}),\ \texttt{M}_1(\mathrm{x},\mathrm{y}_\mathrm{w})))$ $\mathsf{P_H} := \texttt{Or}(\texttt{And}(\texttt{Ref}(\mathrm{x},\mathrm{y}),\texttt{Ref}_1(\mathrm{x},\mathrm{y}_\mathrm{w}),\texttt{M}(\mathrm{x},\mathrm{y}_\mathrm{l})),\ \texttt{And}(\texttt{Ref}(\mathrm{x},\mathrm{y}_\mathrm{l}),\texttt{M}(\mathrm{x},\mathrm{y}_\mathrm{w}),\ \texttt{M}_1(\mathrm{x},\mathrm{y}_\mathrm{w})))$

If the reference model deems the winner to be good and our model says that the loser is good, then our model should deem the winner to be good ~~and the reference model should deem the loser is good.~~

Losses with a reference model

SimPO	P := Implies(And(Mref(x,y_w),M(x,y_l)), And(Mref(x,y_l),M(x,y_w))) P_H := Or(And(Mref(x,y_l),M(x,y_w)), And(Mref(x,y_w),M(x,y_l)))
DPO	P := Implies(And(Ref(x,y_w),M(x,y_l)), ~~And(Ref(x,y_l),M(x,y_w))~~) P_H := Or(~~And(Ref(x,y_l),M(x,y_w))~~, And(Ref(x,y_w),M(x,y_l)))
DPOP	P := Implies(And(Ref(x,y),Ref_1(x,y_w),M(x,y_l)), And(Ref(x,y_l),M(x,y_w), M_1(x,y_w))) P_H := Or(And(Ref(x,y),Ref_1(x,y_w),M(x,y_l)), And(Ref(x,y_l),M(x,y_w), M_1(x,y_w)))

If the reference model deems the winner to be good and our model says that the loser is good, then our model should deem the winner to be good ~~and the reference model should deem the loser is good.~~

$$-\log\sigma\left(\log\frac{\pi_{\text{ref}}(y_l \mid x)\pi_\theta(y_w \mid x)(1-\pi_\theta(y_l \mid x))}{\pi_\theta(y_l \mid x)\pi_{\text{ref}}(y_w \mid x)(1-\pi_\theta(y_w \mid x))}\right)$$

- We can play with the semantics to define new classes of DPO losses, currently exploring this empirically.

Losses with a reference model

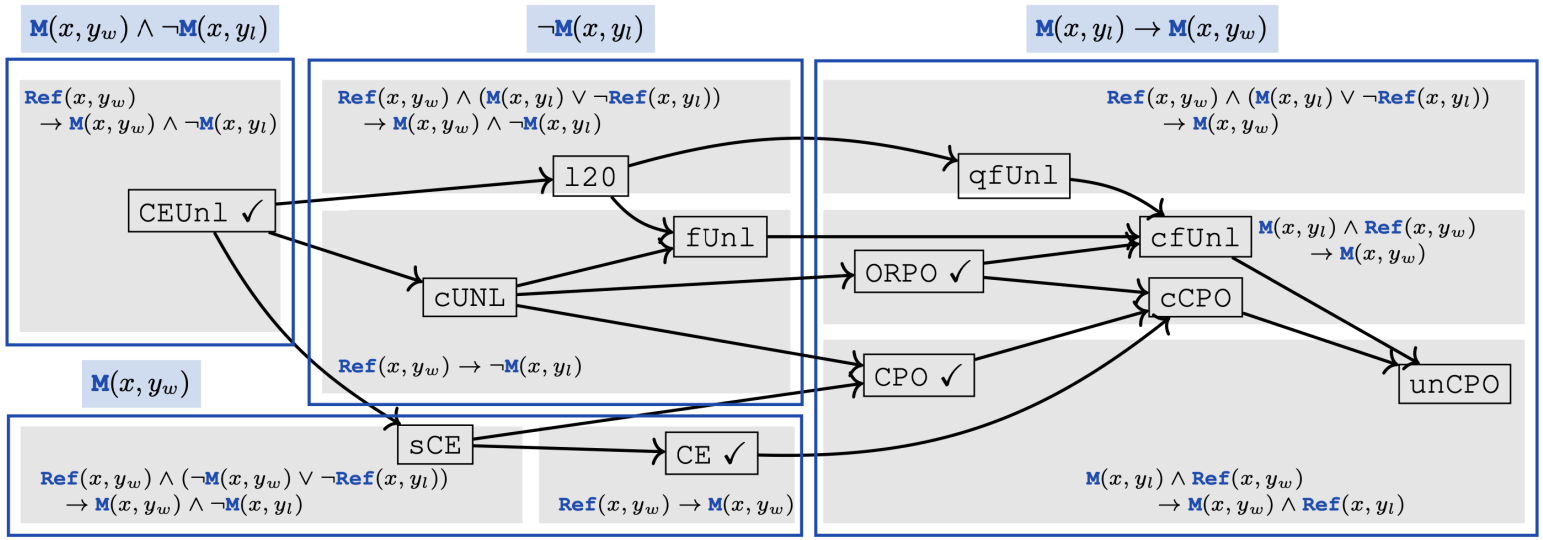
SimPO	P := Implies(And(Mref(x,y_w),M(x,y_l)), And(Mref(x,y_l),M(x,y_w))) P_H := Or(And(Mref(x,y_l),M(x,y_w)), And(Mref(x,y_w),M(x,y_l)))
DPO	P := Implies(And(Ref(x,y_w),M(x,y_l)), ~~And(Ref(x,y_l),M(x,y_w))~~) P_H := Or(~~And(Ref(x,y_l),M(x,y_w))~~, And(Ref(x,y_w),M(x,y_l)))
DPOP	P := Implies(And(Ref(x,y),Ref_1(x,y_w),M(x,y_l)), And(Ref(x,y_l),M(x,y_w), M_1(x,y_w))) P_H := Or(And(Ref(x,y),Ref_1(x,y_w),M(x,y_l)), And(Ref(x,y_l),M(x,y_w), M_1(x,y_w)))

If the reference model deems the winner to be good and our model says that the loser is good, then our model should deem the winner to be good ~~and the reference model should deem the loser is good.~~

$$-\log\sigma\left(\log\frac{\pi_{\text{ref}}(y_l \mid x)\pi_\theta(y_w \mid x)(1-\pi_\theta(y_l \mid x))}{\pi_\theta(y_l \mid x)\pi_{\text{ref}}(y_w \mid x)(1-\pi_\theta(y_w \mid x))}\right)$$

strategy (from before): make DPO more/less constrained, and follow which directions look empirically promising, refine.

The reference loss landscape



- Systematic procedure for generating reference losses from single model losses, much to explore.

Conclusions

- New ideas about formalizing preference loss functions using symbolic techniques, developed new technical tools for this.

Conclusions

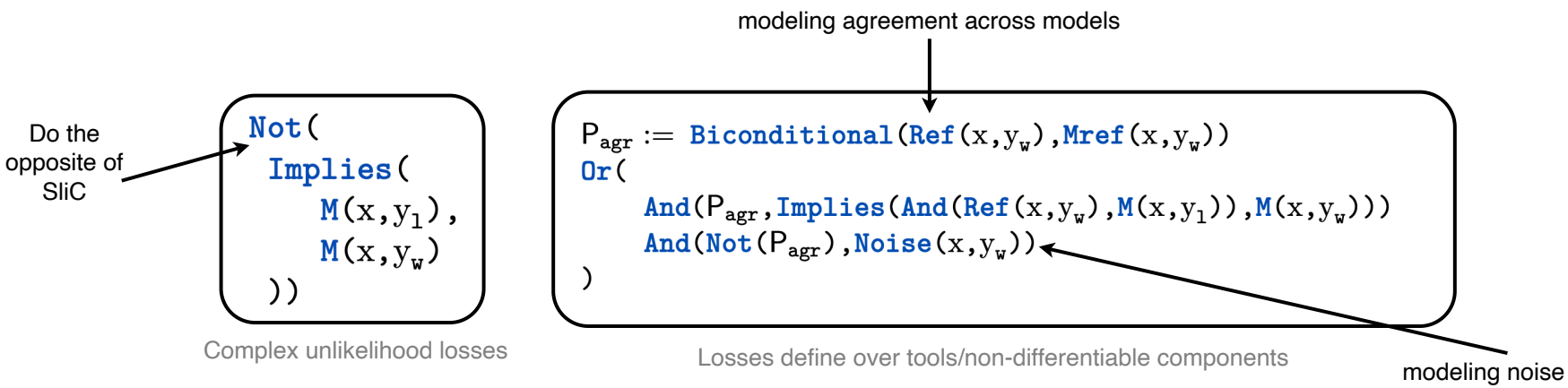
- New ideas about formalizing preference loss functions using symbolic techniques, developed new technical tools for this.

  The procedure: write a (high-level) symbolic program, or modify an existing one, compile into a loss and experiment (then repeat).

Conclusions

- New ideas about formalizing preference loss functions using symbolic techniques, developed new technical tools for this.

  The procedure: write a (high-level) symbolic program, or modify an existing one, compile into a loss and experiment (then repeat).



- Recipe for building much more complex reasoning systems with and on top of LLMs.

Thank you.

References I

Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. (2023). A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*.

Bogin, B., Yang, K., Gupta, S., Richardson, K., Bransom, E., Clark, P., Sabharwal, A., and Khot, T. (2024). Super: Evaluating agents on setting up and executing tasks from research repositories. *arXiv preprint arXiv:2409.07440*.

Chavira, M. and Darwiche, A. (2008). On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799.

Chen, J., Yuan, S., Ye, R., Majumder, B. P., and Richardson, K. (2023). Put your money where your mouth is: Evaluating strategic planning and execution of llm agents in an auction arena. *arXiv preprint arXiv:2310.05746*.

Dai, J., Pan, X., Sun, R., Ji, J., Xu, X., Liu, M., Wang, Y., and Yang, Y. (2024). Safe rlhf: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*.

Gu, Y., Mishra, B. D., and Clark, P. (2023). Do language models have coherent mental models of everyday things? *Proceedings of ACL*.

Hong, J., Lee, N., and Thorne, J. (2024). Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*.

Jeffrey, R. C. (1965). *The logic of decision*. University of Chicago press.

References II

Ji, J., Liu, M., Dai, J., Pan, X., Zhang, C., Bian, C., Chen, B., Sun, R., Wang, Y., and Yang, Y. (2024). Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Kassner, N., Tafjord, O., Sabharwal, A., Richardson, K., Schuetze, H., and Clark, P. (2023). Language models with rationality. *Proceedings of EMNLP*.

Li, T., Gupta, V., Mehta, M., and Srikumar, V. (2019). A Logic-Driven Framework for Consistency of Neural Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3924–3935, Hong Kong, China. Association for Computational Linguistics.

Meng, Y., Xia, M., and Chen, D. (2024). Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Pal, A., Karkhanis, D., Dooley, S., Roberts, M., Naidu, S., and White, C. (2024). Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*.

Park, R., Rafailov, R., Ermon, S., and Finn, C. (2024). Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. (2024). Direct preference optimization: Your language model is secretly a reward model. *Proceedings of Neurips*.

Rescher, N. (1967). *The logic of decision and action*. University of Pittsburgh Pre.

References III

Tang, Y., Guo, Z. D., Zheng, Z., Calandriello, D., Munos, R., Rowland, M., Richemond, P. H., Valko, M., Pires, B. Á., and Piot, B. (2024). Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. (2019). Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Xu, J., Zhang, Z., Friedman, T., Liang, Y., and Broeck, G. (2018). A Semantic Loss Function for Deep Learning with Symbolic Knowledge. In *International Conference on Machine Learning*, pages 5498–5507.

Yang, R., Chen, J., Zhang, Y., Yuan, S., Chen, A., Richardson, K., Xiao, Y., and Yang, D. (2024). Selfgoal: Your language agents already know how to achieve high-level goals. *arXiv preprint arXiv:2406.04784*.

Zhang, Y., Yuan, S., Hu, C., Richardson, K., Xiao, Y., and Chen, J. (2024). Timearena: Shaping efficient multitasking language agents in a time-aware simulation. *arXiv preprint arXiv:2402.05733*.

Zhao, Y., Khalman, M., Joshi, R., Narayan, S., Saleh, M., and Liu, P. J. (2022). Calibrating sequence likelihood improves conditional language generation. In *The eleventh international conference on learning representations*.