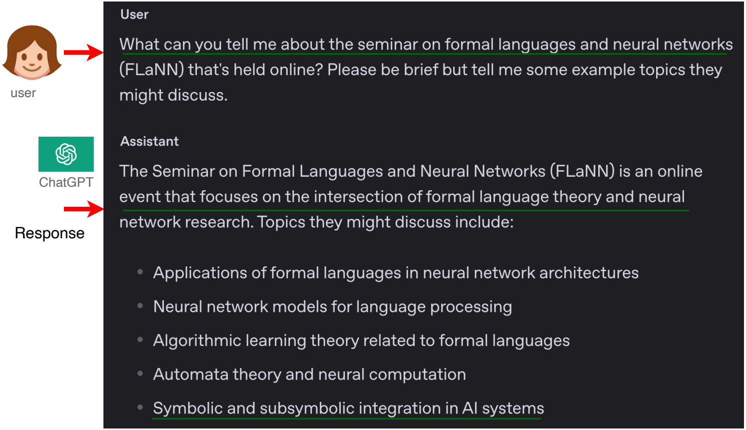
Understanding the Logic of Direct Preference Alignment through Logic

Kyle Richardson[1] Vivek Srikumar[2] Ashish Sabharwal[1]

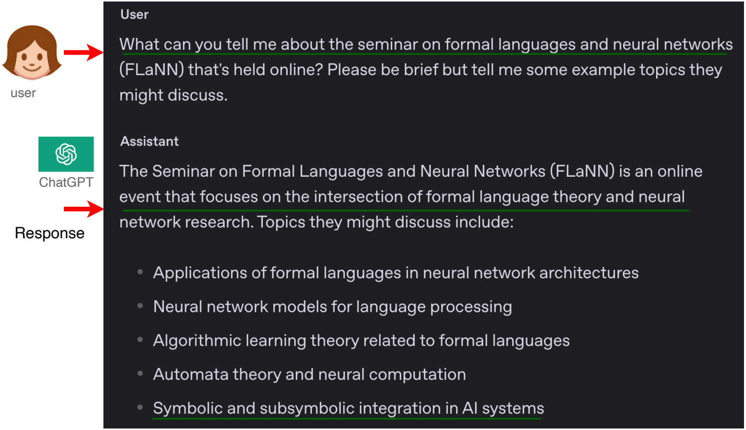
Allen Institute for AI (AI2)[1]
University of Utah[2]

March 2025

General purpose large language models (LLMs)

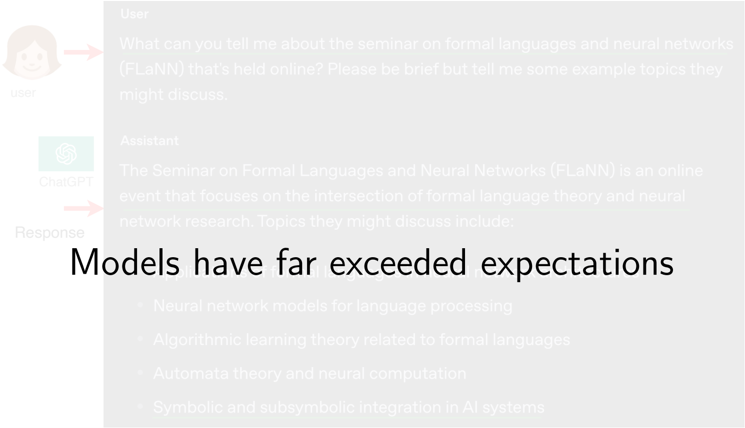


General purpose large language models (LLMs)



- We routinely use LLMs for writing code, integrate within complex systems, use for tasks like scientific discovery.

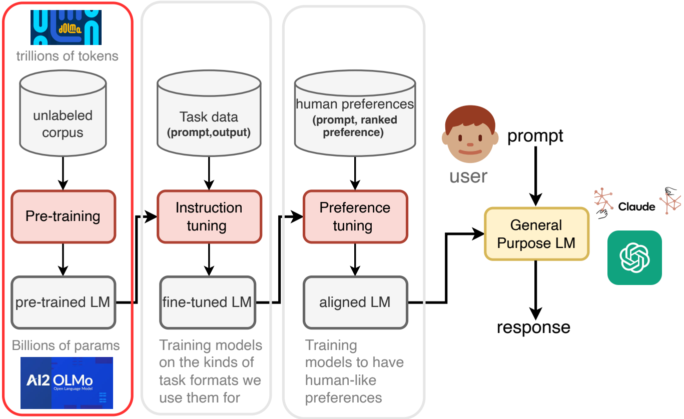
General purpose large language models (LLMs)



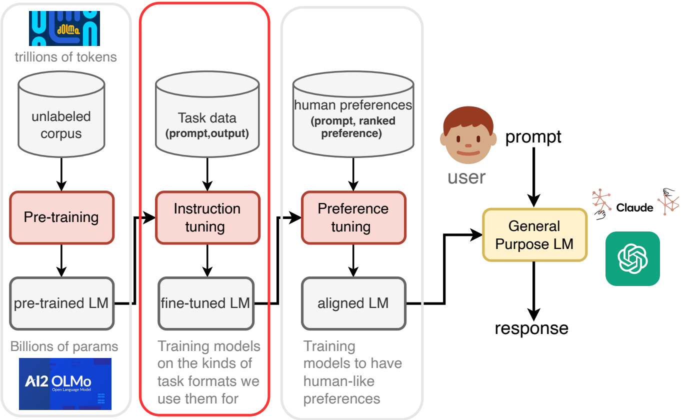
Models have far exceeded expectations

- We routinely use LLMs for writing code, integrate within complex systems, use for tasks like scientific discovery.

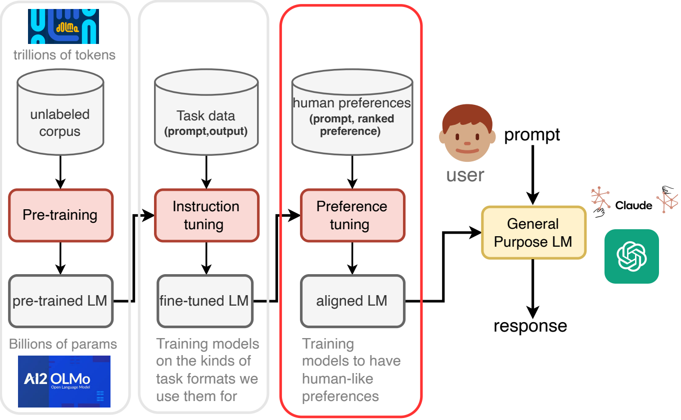
How do we get to general purpose LLMs? **reminder**



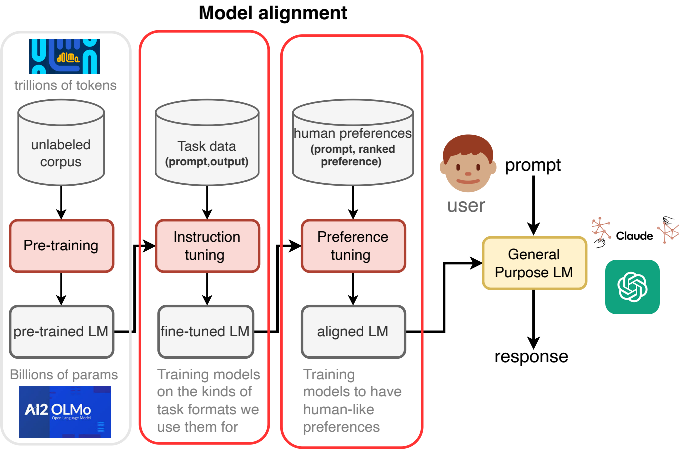
How do we get to general purpose LLMs? **reminder**



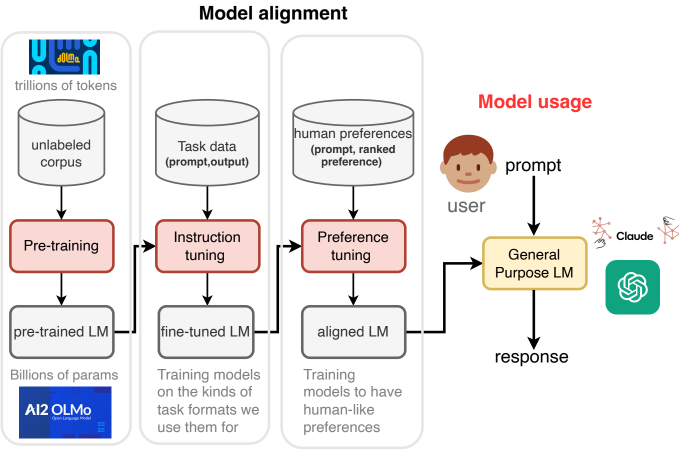
How do we get to general purpose LLMs? **reminder**



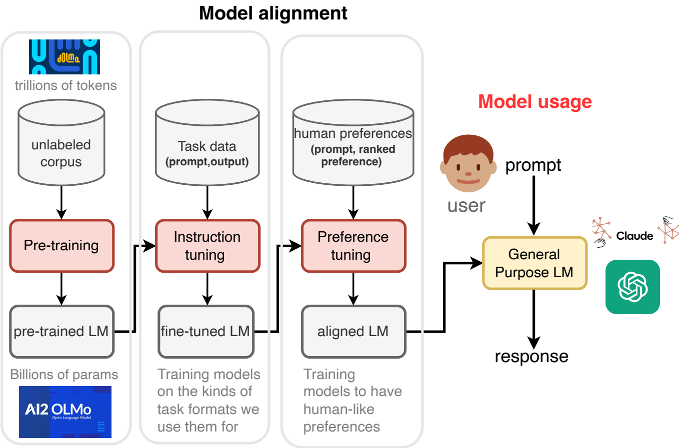
How do we get to general purpose LLMs? **reminder**



How do we get to general purpose LLMs? **reminder**

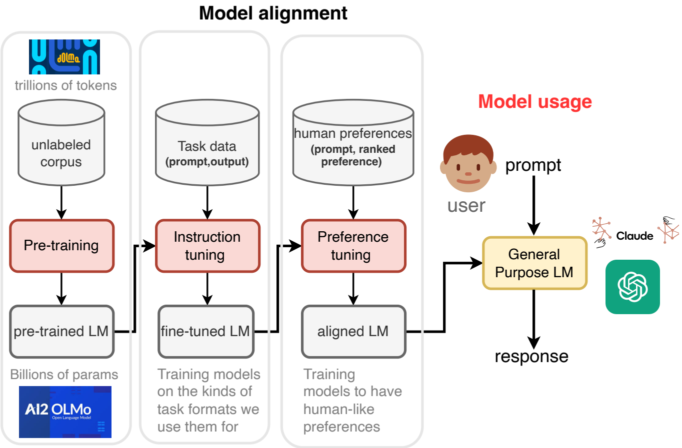


How do we get to general purpose LLMs? **reminder**



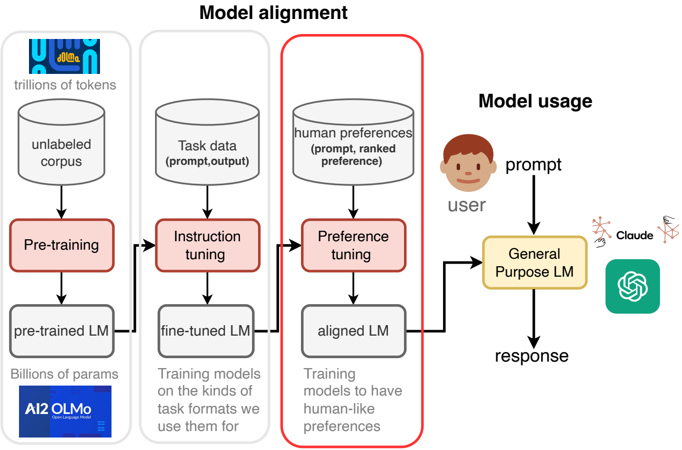
- **Dilemma**: we know vanishingly little about commercial models, models and datasets in general are huge, opaque.

How do we get to general purpose LLMs? **reminder**



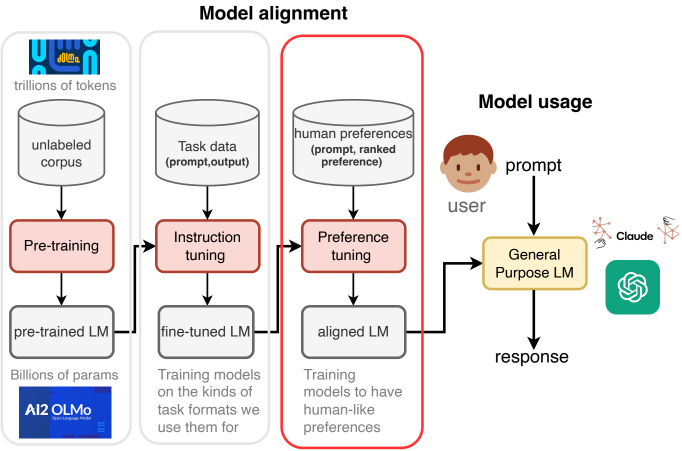
An obvious problem for safety and applications, but also for deciding what research to do, how to innovate.

Modeling the formal semantics of LLM algorithms



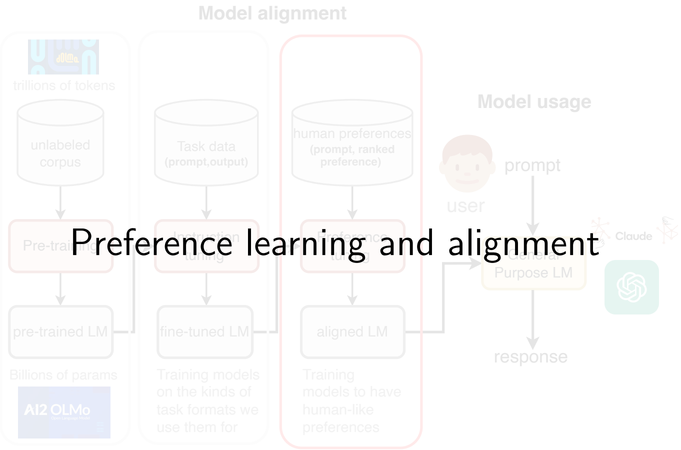
Today: can we formally characterize the semantics of preference tuning and alignment? Both for understanding and innovation; **armchair NLP**.

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?

Modeling the formal semantics of LLM algorithms



Preference learning and alignment

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

Offline preference alignment in a nutshell

- Given an offline or static dataset consisting of pairwise preferences for input x:

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

optimize a policy model $y \sim \pi_\theta(\cdot \mid x)$ (**LLM**) to such preferences.

Offline preference alignment in a nutshell

- Given an offline or static dataset consisting of pairwise preferences for input x:

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

optimize a policy model $y \sim \pi_\theta(\cdot \mid x)$ (**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, takes the form (Tang et al., 2024):

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

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function, takes the form (Tang et al., 2024):

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

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function, takes the form (Tang et al., 2024):

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

model quantity

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function, takes the form (Tang et al., 2024):

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

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

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

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function, takes the form (Tang et al., 2024):

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

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

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

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function, takes the form (Tang et al., 2024):

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

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

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

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, takes the form (Tang et al., 2024):

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

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

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

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, takes the form (Tang et al., 2024):

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

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

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

Direct preference alignment (DPA) algorithms

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function, takes the form (Tang et al., 2024):

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

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

	$f(\rho_\theta, \beta) =$	ρ_θ	**properties**
`DPO`	$-\log \sigma(\beta \rho_\theta)$	$\log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}$	logistic log loss
`IPO`	$(\rho_\theta - \frac{1}{2\beta})^2$		squared loss
`SliC`	$\max(0, \beta - \rho_\theta)$	$\log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}$	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, takes the form (Tang et al., 2024):

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

The procedure: Select a convex loss function f, define some model quantity ρ_θ, experiment.

Coming up with a new preference loss

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function, takes the form (Tang et al., 2024):

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

The procedure: Select a convex loss function f, define some model quantity ρ_θ, experiment.

1. (theory) *What is the right f to use?* Theoretical limitations and properties of Bradley-Terry model or other variants.

Coming up with a new preference loss

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function, takes the form (Tang et al., 2024):

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

The procedure: Select a convex loss function f, define some model quantity ρ_θ, experiment.

1. (theory) *What is the right f to use?* Theoretical limitations and properties of Bradley-Terry model or other variants.
2. (empirical) *Can we devise novel algorithmic 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, takes the form (Tang et al., 2024):

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

The procedure: Select a convex loss function f, define some model quantity ρ_θ, experiment.

1. (theory) *What is the right f to use?* Theoretical limitations and properties of Bradley-Terry model or other variants.
2. (empirical) *Can we devise novel algorithmic 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?

Coming up with a new preference loss

- Recent direct preference alignment (DPA) approaches assume a closed-form loss function, takes the form (Tang et al., 2024):

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

Some observations about variations of DPO

The procedure: Select a convex loss function f, define some model quantity ρ_θ, experiment.

1. (theory) *What is the right f to use?* Theoretical limitations and properties of Bradley-Terry model or other variants.
2. (empirical) *Can we devise novel algorithmic 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?

A look at the structure of DPO loss functions

- Looking at different preference approaches through time.

A look at the structure of DPO loss functions

- Looking at different preference approaches through time.

Approach	Loss equation $f(\rho_\theta, \beta)$
SliC (Zhao et al., 2022)	$\max\left(0, \beta - \log\frac{\pi_\theta(y_w\|x)}{\pi_\theta(y_l\|x)}\right)$

A look at the structure of DPO loss functions

- Looking at different preference approaches through time.

Approach	Loss equation $f(\rho_\theta, \beta)$
SliC (Zhao et al., 2022)	$\max\left(0, \beta - \log \frac{\pi_\theta(y_w\|x)}{\pi_\theta(y_l\|x)}\right)$
CPO (Xu et al., 2024)	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w\|x)}{\pi_\theta(y_l\|x)}\right)$

A look at the structure of DPO loss functions

- Looking at different preference approaches through time.

Approach	Loss equation $f(\rho_\theta, \beta)$
SliC (Zhao et al., 2022)	$\max\left(0, \beta - \log \frac{\pi_\theta(y_w\|x)}{\pi_\theta(y_l\|x)}\right)$
CPO (Xu et al., 2024)	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w\|x)}{\pi_\theta(y_l\|x)}\right)$

A look at the structure of DPO loss functions

- Looking at different preference approaches through time.

Approach	**Loss equation** $f(\rho_\theta, \beta)$
SliC (Zhao et al., 2022)	$\max\left(0, \beta - \log\frac{\pi_\theta(y_w\|x)}{\pi_\theta(y_l\|x)}\right)$
CPO (Xu et al., 2024)	$-\log\sigma\left(\beta\log\frac{\pi_\theta(y_w\|x)}{\pi_\theta(y_l\|x)}\right)$

Remove optimization details, log ratio $s_\theta(y_w, y_l) := \log\frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}$, *regularizer terms, length normalization*, **core loss equations** ρ_θ.

A look at the structure of DPO loss functions

- Looking at different preference approaches through time.

Approach	**Loss equation** $f(\rho_\theta, \beta)$
SliC (Zhao et al., 2022)	$\max\left(0, \beta - \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$
CPO (Xu et al., 2024)	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$
RRHF (Yuan et al., 2024)	$\max\left(0, -\log \frac{\pi_\theta(y_w \mid x)^{\mid y_w \mid}}{\pi_\theta(y_l \mid x)^{\mid y_l \mid}}\right)$

note: This is a messy area, idiosyncratic.

Remove optimization details, log ratio $s_\theta(y_w, y_l) := \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}$, *regularizer terms, length normalization*, **core loss equations** ρ_θ.

A look at the structure of DPO loss functions

- Looking at different preference approaches through time.

Approach	Loss equation $f(\rho_\theta, \beta)$
SliC (Zhao et al., 2022)	$\max\left(0, \beta - \log \frac{\pi_\theta(y_w\|x)}{\pi_\theta(y_l\|x)}\right)$
CPO (Xu et al., 2024)	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w\|x)}{\pi_\theta(y_l\|x)}\right)$
RRHF (Yuan et al., 2024)	$\max\left(0, -\log \frac{\pi_\theta(y_w\|x)^{\|y_w\|}}{\pi_\theta(y_l\|x)^{\|y_l\|}}\right)$

Remove optimization details, log ratio $s_\theta(y_w, y_l) := \log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}$, *regularizer terms, length normalization*, **core loss equations** ρ_θ.

A look at the structure of DPO loss functions

- Looking at different preference approaches through time.

Approach	Loss equation $f(\rho_\theta, \beta)$
SliC (Zhao et al., 2022)	$\max\left(0, \beta - \log \frac{\pi_\theta(y_w\|x)}{\pi_\theta(y_l\|x)}\right)$
CPO (Xu et al., 2024)	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w\|x)}{\pi_\theta(y_l\|x)}\right)$
RRHF (Yuan et al., 2024)	$\max\left(0, -\log \frac{\pi_\theta(y_w\|x)^{\|y_w\|}}{\pi_\theta(y_l\|x)^{\|y_l\|}}\right)$

observation: same log ratios keep coming up

Remove optimization details, log ratio $s_\theta(y_w, y_l) := \log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}$, *regularizer terms, length normalization*, **core loss equations** ρ_θ.

A look at the structure of DPO loss functions

- Looking at different preference approaches through time.

Approach	Loss equation $f(\rho_\theta, \beta)$
SliC (Zhao et al., 2022)	$\max\left(0, \beta - \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$
CPO (Xu et al., 2024)	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$
RRHF (Yuan et al., 2024)	$\max\left(0, -\log \frac{\pi_\theta(y_w \mid x)^{\mid y_w \mid}}{\pi_\theta(y_l \mid x)^{\mid y_l \mid}}\right)$

question: What do these log ratios mean semantically?

Remove optimization details, log ratio $s_\theta(y_w, y_l) := \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}$, *regularizer terms, length normalization*, **core loss equations** ρ_θ.

A look at the structure of DPO loss functions

- Looking at different preference approaches through time.

Approach	Loss equation $f(\rho_\theta, \beta)$
SliC (Zhao et al., 2022)	$\max\left(0, \beta - \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)$
CPO (Xu et al., 2024)	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right.$
RRHF (Yuan et al., 2024)	$\max\left(0, -\log \frac{\pi_\theta(y_w \mid x)^{\mid y_w \mid}}{\pi_\theta(y_l \mid x)^{\mid y_l \mid}}\right)$

Adding a reference model, involves adding an additional term:

DPO

$$-\log \sigma\left(\beta \cdot \left[\underbrace{\log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}}_{s_\theta(y_w, y_l)} - \underbrace{\log \frac{\pi_{\text{ref}}(y_w \mid x)}{\pi_{\text{ref}}(y_l \mid x)}}_{\textbf{reference policy}}\right]\right)$$

The recipe for creating new variants of DPO

DPO

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

The recipe for creating new variants of DPO

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

- **Type 1**: Add additional terms to loss.

Variant	Core Loss equation ρ_θ
R-DPO (Park et al., 2024)	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l) + \gamma_{\text{length}}$
ODPO (Amini et al., 2024)	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l) - \gamma_{\text{offset}}$
DPOP (Pal et al., 2024)	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l) - \log\frac{\pi_{\text{ref}}(x,y_w)}{\pi_\theta(x,y_w)}$

The recipe for creating new variants of DPO

DPO

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

- **Type 1**: Add additional terms to loss.

Variant	Core Loss equation ρ_θ
R-DPO (Park et al., 2024)	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l) + \gamma_{\text{length}}$
ODPO (Amini et al., 2024)	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l) - \gamma_{\text{offset}}$
DPOP (Pal et al., 2024)	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l) - \log \frac{\pi_{\text{ref}}(x, y_w)}{\pi_\theta(x, y_w)}$

- **Type 2**: Change or re-parameterize reference ratio.

Variant	Core Loss equation ρ_θ
ORPO (Hong et al., 2024)	$s_\theta(y_w, y_l) - \log \frac{1-\pi_\theta(y_w \mid x)}{1-\pi_\theta(y_l \mid x)}$
SimPO (Meng et al., 2024)	$s_\theta(y_w, y_l) - \gamma_{\text{penalty}}$

The recipe for creating new variants of DPO

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

- **Type 1**: Add additional terms to loss.

Variant	Core Loss equation ρ_θ
[illegible] (Park et al., 2024)	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l) - \gamma_{\text{length}}$
ODPO (Amini et al., 2024)	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l) - \gamma_{\text{offset}}$
DPOP (Pal et al., 2024)	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l) - \log \frac{\pi_{\text{ref}}(x, y_w)}{\pi_\theta(x, y_w)}$

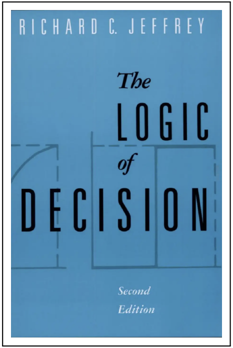
question: What happens when we add new terms?

- **Type 2**: Change or re-parameterize reference ratio.

Variant	Core Loss equation ρ_θ
ORPO (Hong et al., 2024)	$s_\theta(y_w, y_l) - \log \frac{1 - \pi_\theta(y_w \mid x)}{1 - \pi_\theta(y_l \mid x)}$
SimPO (Meng et al., 2024)	$s_\theta(y_w, y_l) - \gamma_{\text{penalty}}$

Haven't these semantic questions been looked at before?

Analytic philosophy: Much work on the semantics of pairwise preference, rich languages for expressing ideas.



(Jeffrey, 1965)

THE STATUS OF VARIOUS PREFERENCE PRINCIPLES

Preference Principle	Von Wright	Chisholm Sosa	Martin	$P^{\sharp}$	$P^{\star}$	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. $[\sim(pPq) \,\&\, \sim(qPr)] \rightarrow \sim(pPr)$		✓		+	+	—
13. $(pPr \lor qPr) \rightarrow (p \lor q)\, Pr$			✓	—	—	—
14. $(p \lor q)\, Pr \rightarrow [pPr \,\&\, qPr]$	✓			—	—	—
15. $[pPr \,\&\, qPr] \rightarrow (p \lor q)\, Pr$	✓			—	—	—
16. $(p \lor q)\, Pr \rightarrow (pPr \lor qPr)$				—	—	—
17. $pP(q \lor r) \rightarrow (pPq \,\&\, pPr)$			✓	—	—	—
18. $(pPq \,\&\, pPr) \rightarrow pP(q \lor r)$				—	—	—
19. $(pPr \,\&\, qPr) \rightarrow (p \,\&\, q)\, Pr$				—	—	—

Semantic foundations for the logic of preference Rescher (1967)

The language of machine learning

Loss functions

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

The language of machine learning

Loss functions

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

- 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

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

↑

Specification or theory of preference?

- 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

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

↑

Specification or theory of preference?

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

Broader goal: High-level modeling languages for specifying and better understanding LLMs and their algorithms.

The language of machine learning

Loss functions

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

↑

Specification or theory of preference?

Formalization of preference losses

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

Broader goal: High-level modeling languages for specifying and better understanding LLMs and their algorithms.

Preference learning as a discrete reasoning problem

Loss Function

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

Preference learning as a discrete reasoning problem

Loss Function

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

Two models, four predictions

Preference learning as a discrete reasoning problem

Loss Function

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

Two models, four predictions

- **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

Symbolic Program

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

Loss Function

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

High-level model behavior

- **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

Symbolic Program

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

Loss Function

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

Decompilation ⟷ Compilation

- **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

Symbolic Program

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

Loss Function

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

Decompilation ⟵⟶ Compilation

- **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

Symbolic Program

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

Loss Function

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

Decompilation ⟷ Compilation

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

  1. **Compilation**: Translating specifications into loss, well studied.

Preference learning as a discrete reasoning problem

Symbolic Program

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

Loss Function

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

Decompilation ⟷ Compilation

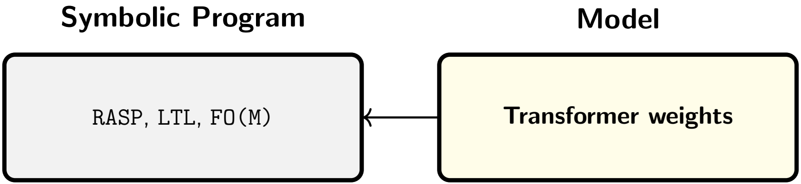
- **Problem:** Given some loss function, can we derive a symbolic program or expression that characterizes the semantics of that loss?
  1. **Compilation**: Translating specifications into loss, well studied.
  2. **Decompilation**:Losses to specifications (inverse), less explored.

Distilling LLMs to symbolic programs in general

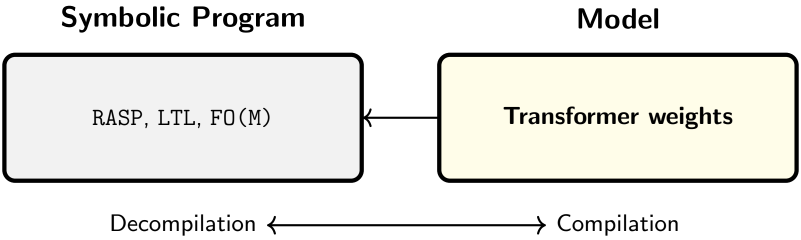
Model

Transformer weights

Distilling LLMs to symbolic programs in general



Distilling LLMs to symbolic programs in general



- We know what the *target languages* are (Weiss et al., 2021; Merrill and Sabharwal, 2023; Yang and Chiang, 2024), how to compile, decompile (Friedman et al., 2023).

What do these programs tell us?

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

What do these programs tell us?

Model predicts loser

Model predicts winner

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

Conceptually: Model predications are logical propositions, Boolean variables inside of formulas, weighted by prediction probability.

What do these programs tell us?

Model predicts loser

Model predicts winner

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

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

Conceptually: Model predications are logical propositions, Boolean variables inside of formulas, weighted by prediction probability.

What do these programs tell us?

Model predicts loser

Model predicts winner

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

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

Conceptually: Predictions are connected through Boolean operators, express constraints on predictions; ρ_θ as formulas.

What do these programs tell us?

Model predicts loser

Model predicts winner

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

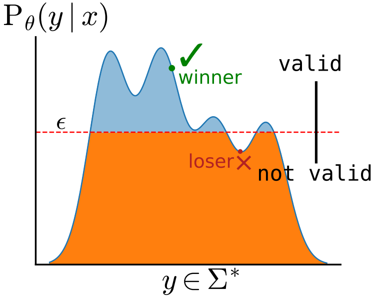
Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

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?

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

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

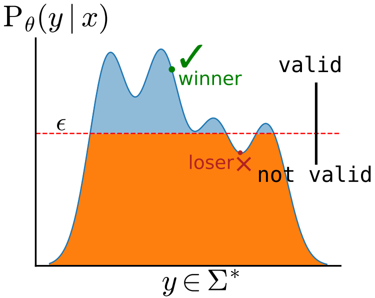


Assumption: Every loss function has an internal logic that can be expressed in this way, we want to uncover that logic.

What do these programs tell us?

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

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

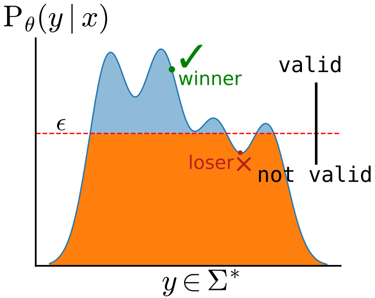


Assumption: Every loss function has an internal logic that can be expressed in this way, we want to uncover that logic.

What do these programs tell us?

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

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



Observation: The second program is more strict than the first, involves semantic entailment.

What do these programs tell us?

Compilation

P

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

$$\ell(D,\theta) = -\log \underbrace{P_\theta(\text{P} \mid D,\theta)}_{\text{probabilistic logic}}$$

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

What do these programs tell us?

Compilation

P

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

$$\ell(D,\theta) = -\log \underbrace{P_\theta(\mathrm{P} \mid D,\theta)}_{\text{probabilistic logic}}$$

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

What we did: defined a novel probabilistic logic for preference modeling,

note: logic useful not only for learning and loss.

What do these programs tell us?

Decompilation

P

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

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

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

What do these programs tell us?

Decompilation

P

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

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

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

$$\underbrace{\ell_{\texttt{CPO}}(D,\theta) = -\log P_\theta(\mathsf{P} \mid D,\theta)}_{\text{correctness property}}$$

What do these programs tell us?

Decompilation

P

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

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

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

$$\underbrace{\ell_{\texttt{CPO}}(D,\theta) = -\log P_\theta(\mathsf{P} \mid D,\theta)}_{\text{correctness property}}$$

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

Illustration of approach and results

Input Loss $\ell_{\mathtt{ORPO}}$

$$-\log\sigma\left(\log\frac{\mathrm{Odds}_\theta(y_w|x)}{\mathrm{Odds}_\theta(y_l|x)}\right)$$

Illustration of approach and results

Input Loss $\ell_{\texttt{ORPO}}$

$$-\log \sigma\left(\log \frac{\text{Odds}_\theta(y_w|x)}{\text{Odds}_\theta(y_l|x)}\right)$$

↓

$$\frac{\rho_\theta^t}{\rho_\theta^b} = \frac{P_\theta(y_w|x)(1-P_\theta(y_l|x))}{P_\theta(y_l|x)(1-P_\theta(y_w|x))}$$

Core loss equation

Illustration of approach and results

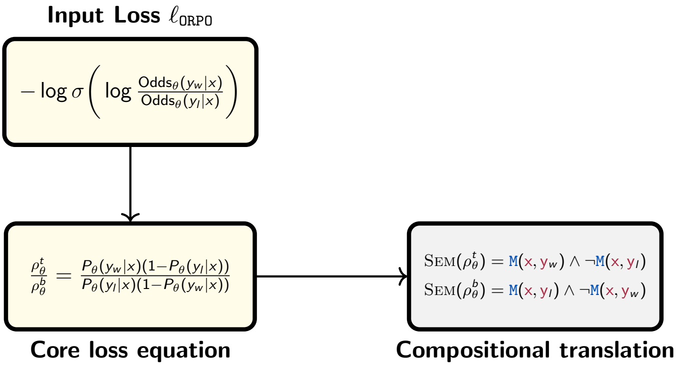


Illustration of approach and results

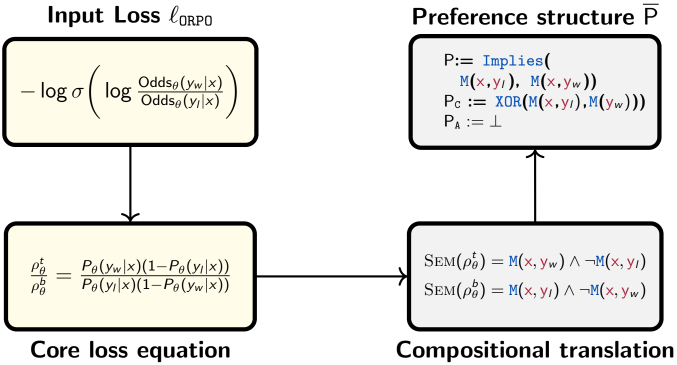


Illustration of approach and results

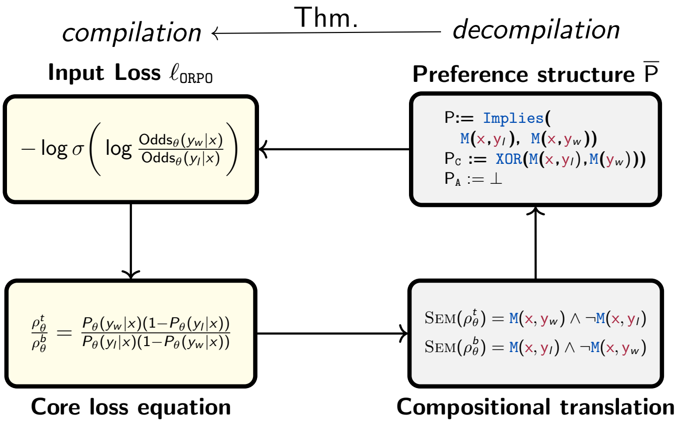
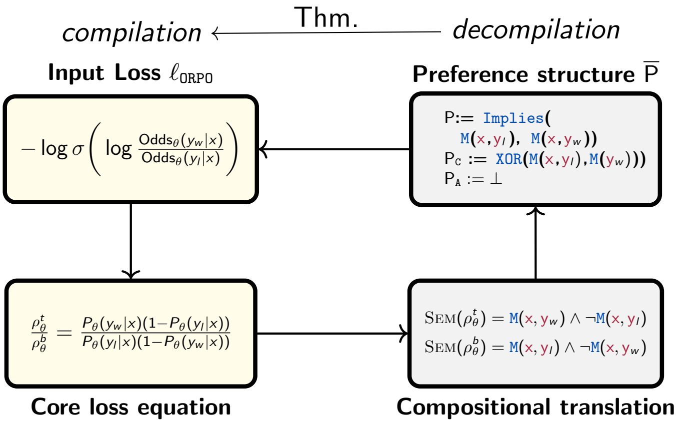
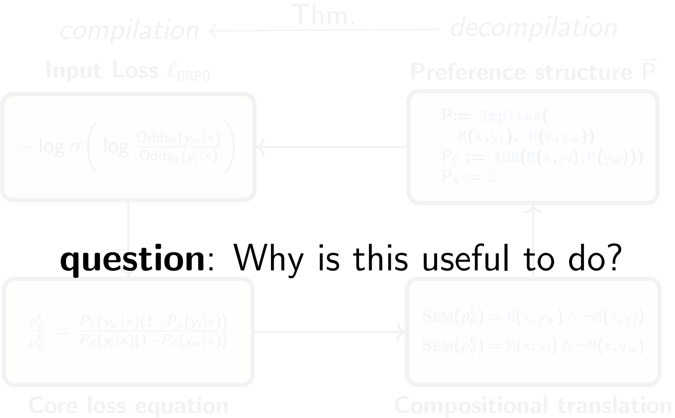


Illustration of approach and results



- **Preference structure**, a core construct in our logic, encoding for preference losses, has a natural Boolean interpretation.

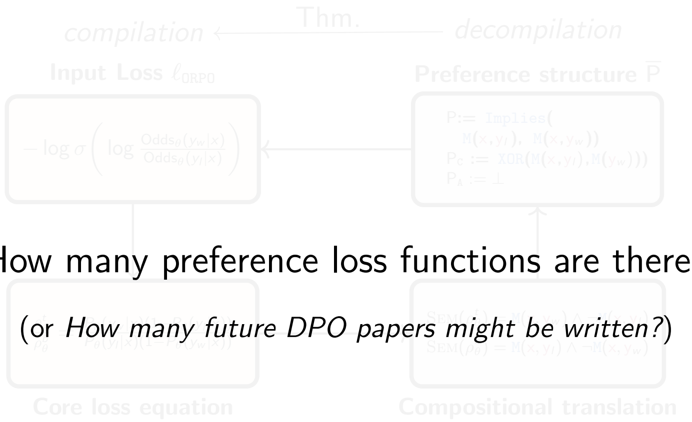
Illustration of approach and results



question: Why is this useful to do?

- **Preference structure**, a core construct in our logic, encoding for preference losses, has a natural Boolean interpretation.

Illustration of approach and results



How many preference loss functions are there?

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

- **Preference structure**, a core construct in our logic, encoding for preference losses, has a natural Boolean interpretation.

Why is this useful? understanding the space

$P^{(1)}$

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

$P^{(2)}$

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

Boolean functions, **2 variables**

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

Why is this useful? understanding the space

$P^{(1)}$

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

$P^{(2)}$

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

Boolean functions, **2 variables**

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

- Every program (in our logic) is pair of Boolean functions (in n variables), corr. to ✓ and ✗, leads to 4^{2^n} possible loss functions.

Why is this useful? understanding the space

$P^{(1)}$

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

$P^{(2)}$

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

Boolean functions, **2 variables**

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

no reference: 256 losses

- Every program (in our logic) is pair of Boolean functions (in n variables), corr. to ✓ and ✗, leads to 4^{2^n} possible loss functions.

Loss functions as truth tables

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

4 variables

Ref(x, y_w)	M(x, y_l)	Ref(x, y_l)	M(x, y_w)
F	F	F	F
F	F	F	T
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

w/ reference: 4,294,967,296 losses

Loss functions as truth tables

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

4 variables

Ref(x, y_w)	M(x, y_l)	Ref(x, y_l)	M(x, y_w)
F	F	F	F
F	F	F	T
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

w/ reference: 4,294,967,296 losses

answer: loads.

Loss functions as truth tables

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

4 variables

Ref(x, y_w)	M(x, y_l)	Ref(x, y_l)	M(x, y_w)
F	F	F	F
F	F	F	T
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

question: How are losses related to one another?

w/ reference: 4,294,967,296 losses

Why is this useful? understanding the structure

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

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

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

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

semantics: $\mathsf{P}^{(2)} \models \mathsf{P}^{(1)}$

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

Proposition (Xu et al., 2018): Loss behavior is monotonic w.r.t semantic entailment: if $\mathsf{P}^{(2)} \models \mathsf{P}^{(1)}$ then $\ell(D, \theta, \mathsf{P}^{(2)}) \geq \ell(D, \theta, \mathsf{P}^{(1)})$.

Why is this useful? understanding the structure

$P^{(1)}$

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

$P^{(2)}$

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

semantics: $P^{(2)} \models P^{(1)}$

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

Proposition (Xu et al., 2018): Loss is equivalent under semantic equivalence: If $P^{(2)} \equiv P^{(1)}$ then $\ell(D, \theta, P^{(2)}) = \ell(D, \theta, P^{(1)})$.

Why is this useful? understanding the structure

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

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

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

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

semantics: $\mathsf{P}^{(2)} \models \mathsf{P}^{(1)}$

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

Theorem: $\ell(D, \theta, \mathsf{P}^{(2)}) > \ell(D, \theta, \mathsf{P}^{(1)})$ (the loss of $\mathsf{P}^{(1)}$ is contained in the loss of $\mathsf{P}^{(2)}$).

Why is this useful? understanding the structure

$P^{(1)}$

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

semantics: $P^{(2)} \models P^{(1)}$

$P^{(2)}$

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

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

answer: Losses are related through their semantics

Theorem: $\ell(D, \theta, P^{(2)}) > \ell(D, \theta, P^{(1)})$ (the loss of $P^{(1)}$ is contained in the loss of $P^{(2)}$).

Why is this useful? understanding the structure

$P^{(1)}$

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

$P^{(2)}$

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

semantics: $P^{(2)} \models P^{(1)}$

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

Practical strategy: Start with empirically successful losses, modify semantics (make more or less constrained), then experiment accordingly.

Why is this useful? understanding the structure

$P^{(1)}$

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

semantics: $P^{(2)} \models P^{(1)}$

$P^{(2)}$

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

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

questions: How does our logic work? What do we see?

Practical strategy: Start with empirically successful losses, modify semantics (make more or less constrained), then experiment accordingly.

How does the logic work? **compilation**

$M(x, y_w)$	$M(x, y_l)$	CPO	ORPO	unCPO
T	T	✓ ✗		✓
T	F	✓	✓	✓
F	T	✗	✗	✗
F	F			✓

P

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

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

How does the logic work? **compilation**

M(x, y_w)	M(x, y_l)	CPO	ORPO	unCPO
T	T	✓ ✗		✓
T	F	✓	✓	✓
F	T	✗	✗	✗
F	F			✓

$$S(i) := \prod_{i \models v} w(v) \cdot \prod_{i \models \neg v} 1 - w(v)$$

P

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

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

How does the logic work? **compilation**

$\texttt{M}(x, y_w)$	$\texttt{M}(x, y_l)$	CPO	ORPO	unCPO
T	T	✓ ✗		✓
T	F	✓	✓	✓
F	T	✗	✗	✗
F	F			✓

P

```
Implies(
  M(x,yl), M(x,yw)
)
```

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

$$S(i) := \prod_{i \models v} w(v) \cdot \prod_{i \models \neg v} 1 - w(v)$$

- Formula probability computed as a weighted count $\sum$ ✓ (Chavira and Darwiche, 2008), loss is $-\log$ (Xu et al., 2018); generalizing:

$$-\log P_\theta(\mathrm{P_x}) := \underbrace{-\log \sigma\left(\log \frac{\sum ✓}{\sum ✗}\right)}_{\text{symmetric to DPA, } \rho_\theta}$$

How does the logic work? **compilation**

M(x, y_w)	M(x, y_l)	CPO	ORPO	unCPO
T	T	✓ ✗		✓
T	F	✓	✓	✓
F	T	✗	✗	✗
F	F			✓

$$S(i) := \prod_{i \models v} w(v) \cdot \prod_{i \models \neg v} 1 - w(v)$$

P

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

Whenever the model deems the loser to be a *valid* *generation, it should deem the winner to be* *valid* *too.*

- Formula probability computed as a weighted count $\sum$ ✓ (Chavira and Darwiche, 2008), loss is $-\log$ (Xu et al., 2018); generalizing:

$$-\log P_\theta(\mathsf{P}_{\texttt{ORPO}}) := -\log \sigma\left(\log \frac{\sum ✓}{\sum ✗}\right)$$

$$= \underbrace{-\log \sigma\left(\log \frac{\pi_\theta(y_w \mid x)(1 - \pi_\theta(y_l \mid x))}{\pi_\theta(y_l \mid x)(1 - \pi_\theta(y_w \mid x))}\right)}_{\ell_{\texttt{ORPO}},\ P_\theta(\mathsf{P}|\textbf{one hot})}$$

How does the logic work? **compilation**

$\texttt{M}(x, y_w)$	$\texttt{M}(x, y_l)$	CPO	ORPO	unCPO
T	T	✓ ✗		✓
T	F	✓	✓	✓
F	T	✗	✗	✗
F	F			✓

P

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

Whenever the model deems the loser to be a <u>valid</u> generation, it should deem the winner to be <u>valid</u> too.

$$S(i) := \prod_{i \models v} w(v) \cdot \prod_{i \models \neg v} 1 - w(v)$$

- Formula probability computed as a weighted count $\sum$ ✓ (Chavira and Darwiche, 2008), loss is $-\log$ (Xu et al., 2018); generalizing:

$$-\log P_\theta(\mathsf{P}_{\texttt{CPO}}) := -\log \sigma \left(\log \frac{\sum ✓}{\sum ✗} \right)$$

$$= \underbrace{-\log \sigma \left(\log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)} \right)}_{\ell_{\texttt{CPO}},\ \sim P_\theta(\mathsf{P} \mid \textbf{one true})}$$

How does the logic work? **compilation**

$\texttt{M}(x, y_w)$	$\texttt{M}(x, y_l)$	CPO	ORPO	unCPO
T	T	✓ ✗		✓
T	F	✓	✓	✓
F	T	✗	✗	✗
F	F			✓

P

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

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

$S(i) := \prod w(v) \cdot \prod 1 - w(v)$

observation: losses differ in conditioning constraints

- Formula probability computed as a weighted count $\sum$ ✓ (Chavira and Darwiche, 2008), loss is $-\log$ (Xu et al., 2018); generalizing:

$$-\log P_\theta(\mathrm{P}_{\texttt{CPO}}) := -\log \sigma\left(\log \frac{\sum ✓}{\sum ✗}\right)$$

$$= \underbrace{-\log \sigma\left(\log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right)}_{\ell_{\texttt{CPO}},\ \sim P_\theta(\mathrm{P} \mid \textbf{one true})}$$

How does the logic work? **compilation**

M(x, y_w)	M(x, y_l)	CPO	ORPO	unCPO
T	T	✓ ✗		✓
T	F	✓	✓	✓
F	T	✗	✗	✗
F	F			✓

P

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

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

$$S(i) := \prod_{i \models v} w(v) \cdot \prod_{i \models \neg v} 1 - w(v)$$

Loss	**Representation $\overline{\mathsf{P}}$**
CE	P := **M**(x,y_w), P_C := ⊥
CEUnl	P := **And**(**M**(x,y_w), **Not**(**M**(x,y_l))) P_C := ⊥
CPO	;; core semantic formula P := **Implies**(**M**(x,y_l), **M**(x,y_w)) ;; one–true constraint P_C := **Or**(**M**(x,y_l), **M**(x,y_w))
ORPO	P := **Implies**(**M**(x,y_l),**M**(x,y_w)) ;; one–hot constraint P_C := **XOR**(**M**(x,y_l), **M**(x,y_w))

How does the logic work? **compilation**

M(x, y_w)	M(x, y_l)	CPO	ORPO	unCPO
T	T	✓ ✗		✓
T	F	✓	✓	✓
F	T	✗	✗	✗
F	F			✓

P

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

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

$$S(i) := \prod_{i \models v} w(v) \cdot \prod_{i \models \neg v} 1 - w(v)$$

- Formula probability computed as a weighted count $\sum$ ✓ (Chavira and Darwiche, 2008), loss is $-\log$ (Xu et al., 2018); generalizing:

$$-\log P_\theta(\mathrm{P}_{\texttt{unCPO}}) := -\log \sigma\left(\log \frac{\sum ✓}{\sum ✗}\right)$$

$$= \underbrace{-\log \sigma\left(\log \frac{\pi_\theta(y_l \mid x)\pi_\theta(y_w \mid x) + (1 - \pi_\theta(y_l \mid x))}{\pi_\theta(y_l \mid x)(1 - \pi_\theta(y_w \mid x))}\right)}_{\textbf{novel loss}}$$

How does the logic work? **compilation**

$\texttt{M}(x, y_w)$	$\texttt{M}(x, y_l)$	CPO	ORPO	unCPO
T	T	✓ ✗		✓
T	F	✓	✓	✓
F	T	✗	✗	✗
F	F			✓

P

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

Whenever the model deems the loser to be a valid generation, it should deem the winner to be valid too.

$S(i) := \prod w(\cdot) \cdot \prod 1 - w(\cdot)$

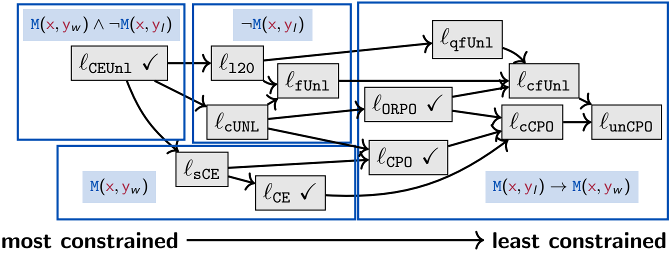
note: $\texttt{M}(x, y_l) \rightarrow \texttt{M}(x, y_w) \equiv \neg\texttt{M}(x, y_l) \vee \texttt{M}(x, y_w)$

- Formula probability computed as a weighted count $\sum$ ✓ (Chavira and Darwiche, 2008), loss is $-\log$ (Xu et al., 2018); generalizing:

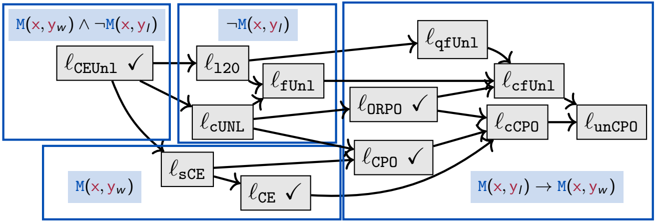
$$-\log P_\theta(\text{P}_{\text{unCPO}}) := -\log \sigma\left(\log \frac{\sum ✓}{\sum ✗}\right)$$

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

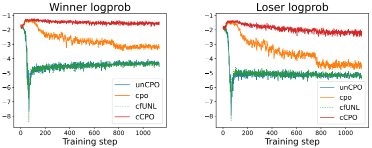
The no reference loss landscape

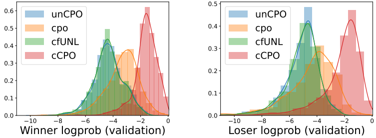


The no reference loss landscape



most constrained ⟶ **least constrained**





Training dynamics **Inference**

The no reference loss landscape

question: Are any of these losses good?

Training dynamics

Inference

Adding a reference model

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

Whenever the model being tuned deems the loser to be a valid generation and the reference model deems the winner to be valid, the tuned model should deem the winner to be valid too, and the reference should deem the loser to be valid.

Adding a reference model

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

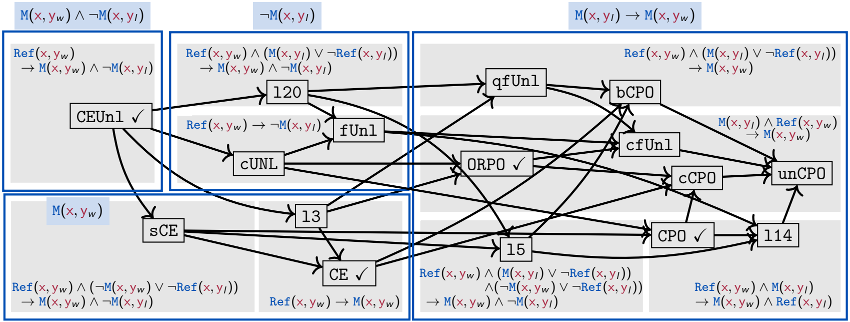
Whenever the model being tuned deems the loser to be a valid generation and the reference model deems the winner to be valid, the tuned model should deem the winner to be valid too, and the reference should deem the loser to be valid.

- **Peculiar semantics**, but the logic makes sense, e.g., we want to maximize

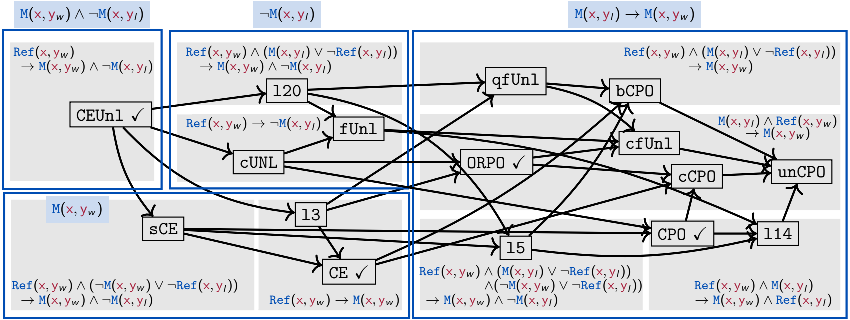
$$\sigma\left(\log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)} - \frac{\pi_{\text{ref}}(y_w \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right)$$

negating left side of implication (i.e., making $\texttt{M}(x, y_l)$ and $\texttt{Ref}(x, y_w)$ false) and making the right side true is logical.

The full landscape, **reference approaches**



The full landscape, **reference approaches**



- Many new losses to explore and experiment with!

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.

  1. Understanding the full space of loss functions (finding: it's a huge space, many novel variations yet to be explored)
  2. Understanding the structure of the space and relationships between different losses (finding: tied to the semantics of the losses).

Conclusions

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

  1. Understanding the full space of loss functions (finding: it's a huge space, many novel variations yet to be explored)
  2. Understanding the structure of the space and relationships between different losses (finding: tied to the semantics of the losses).

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

Thank you.

References I

Amini, A., Vieira, T., and Cotterell, R. (2024). Direct preference optimization with an offset. *arXiv preprint arXiv:2402.10571*.

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*.

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

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*.

Friedman, D., Wettig, A., and Chen, D. (2023). Learning transformer programs. *Advances in Neural Information Processing Systems*, 36:49044–49067.

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.

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.

References II

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

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

Merrill, W. and Sabharwal, A. (2023). A logic for expressing log-precision transformers. *Advances in neural information processing systems*, 36:52453–52463.

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.

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*.

Weiss, G., Goldberg, Y., and Yahav, E. (2021). Thinking like transformers. In *International Conference on Machine Learning*, pages 11080–11090. PMLR.

References III

Xu, H., Sharaf, A., Chen, Y., Tan, W., Shen, L., Van Durme, B., Murray, K., and Kim, Y. J. (2024). Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*.

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, A. and Chiang, D. (2024). Counting like transformers: Compiling temporal counting logic into softmax transformers. *arXiv preprint arXiv:2404.04393*.

Yuan, H., Yuan, Z., Tan, C., Wang, W., Huang, S., and Huang, F. (2024). Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36.

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*.