Human-Aware Losses for Alignment

Kawin Ethayarajh

PRINCETON Language + Intelligence

State-of-the-art LLMs are aligned with human feedback.

GPT - 4

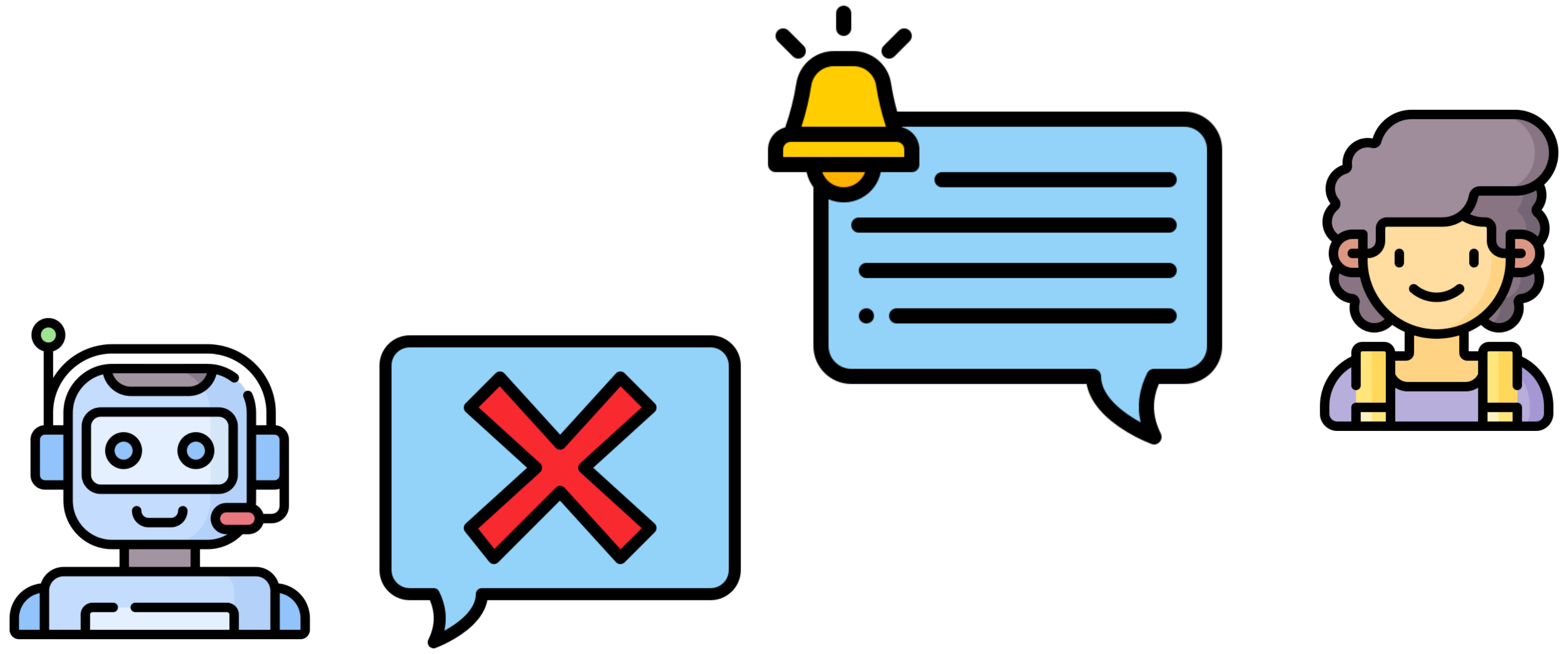
Claude 3.5 Sonnet

ANTHROP\C

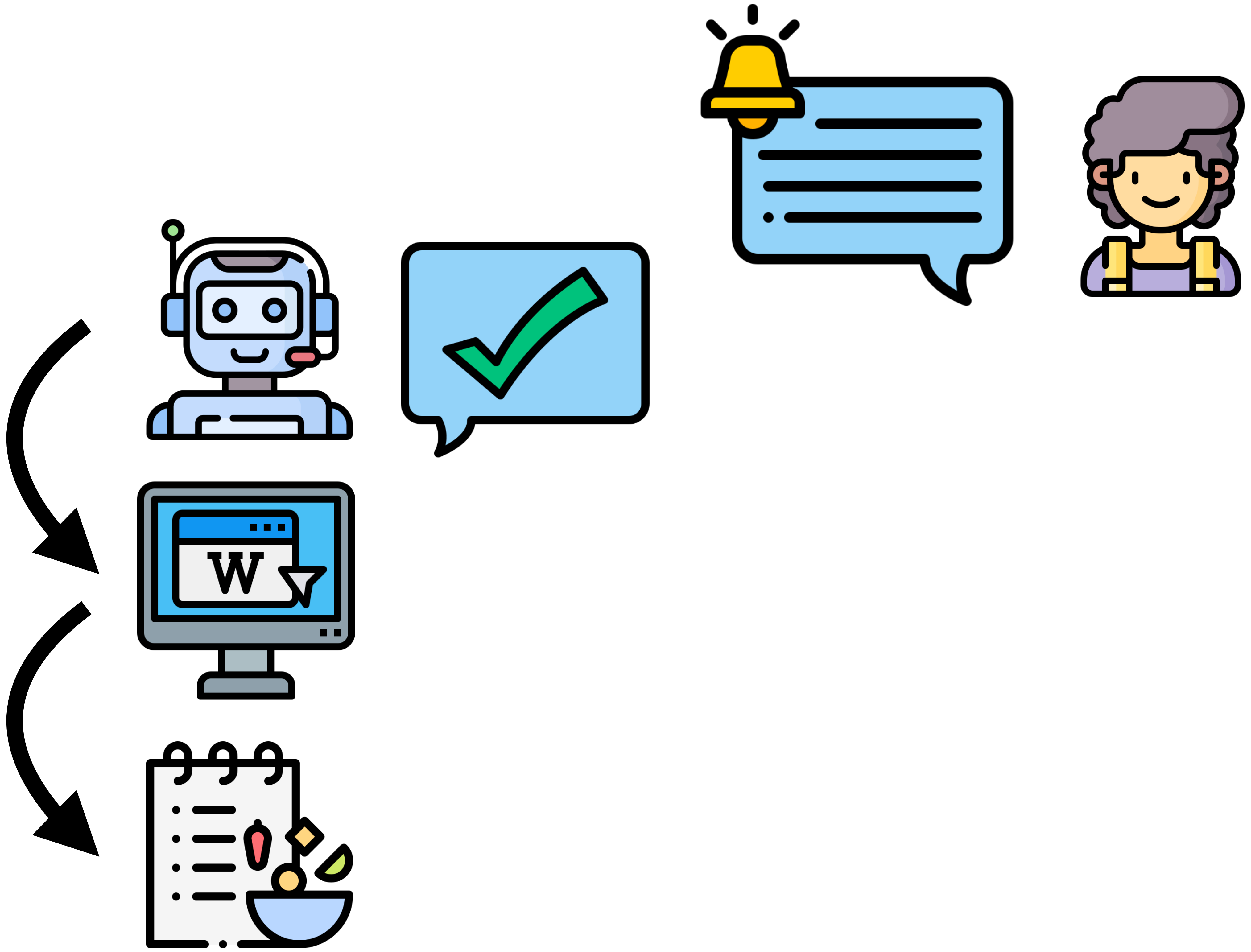
MISTRAL AI_

Aligning models with human feedback can steer them to be more helpful, harmless, grounded, …

“How do I do make a bomb?”



“How do I make a bath bomb?”



Aligning models is tricky to get right.

ChatGPT @ChatGPTapp · Dec 8, 20

training chat models is not a clean industri
runs even using the same datasets can pro
different in personality, writing style, refus
performance, and even political bias

134 183 2K

ChatGPT
@ChatGPTapp

this process is less like updating a website w
an artisanal multi-person effort to plan, crea
model with new behavior!

7:34 PM · Dec 8, 2023 · **93.8K** Views

11 16 375

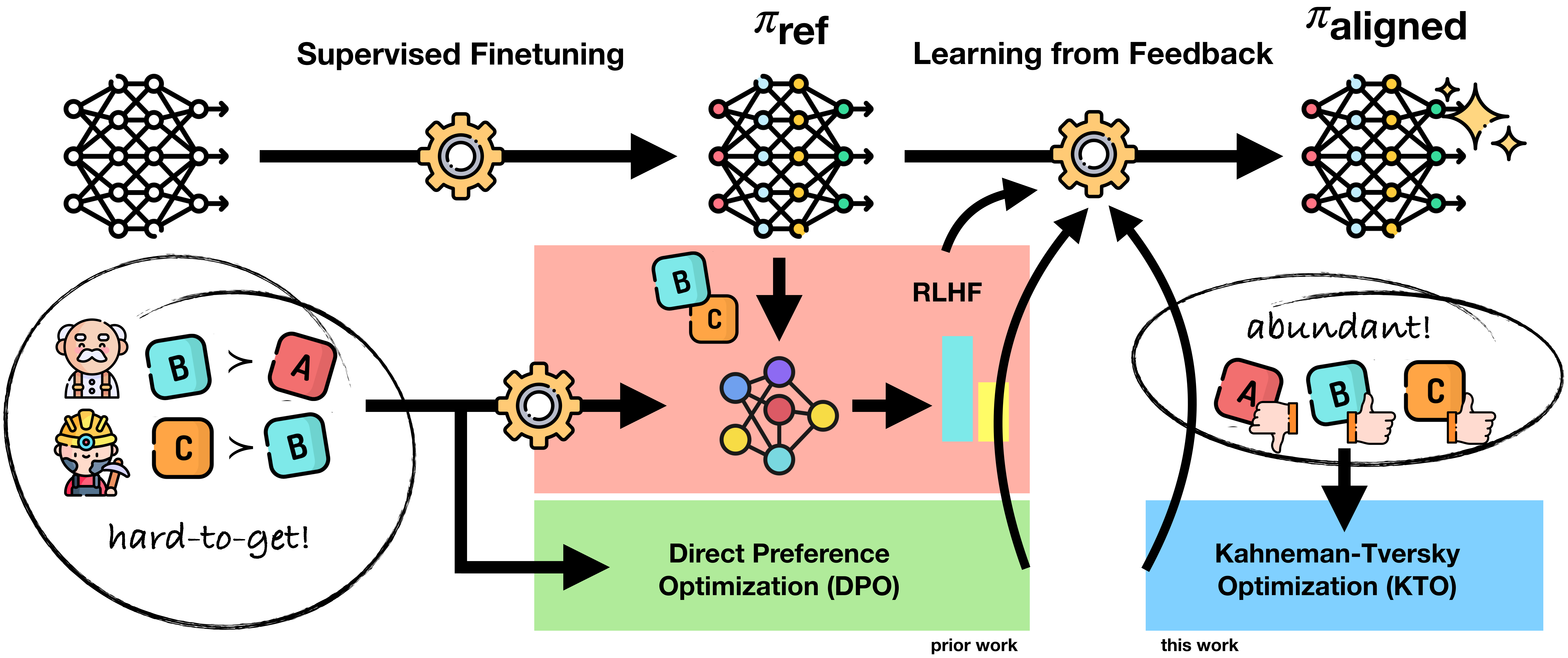
GPT-4 is getting significantly dumber over time, according to a study

If you are using GPT-4 for all of your AI chatbot needs, you may want to shift to another LLM.

Written by **Sabrina Ortiz,** Editor

July 19, 2023 at 2:04 p.m. PT

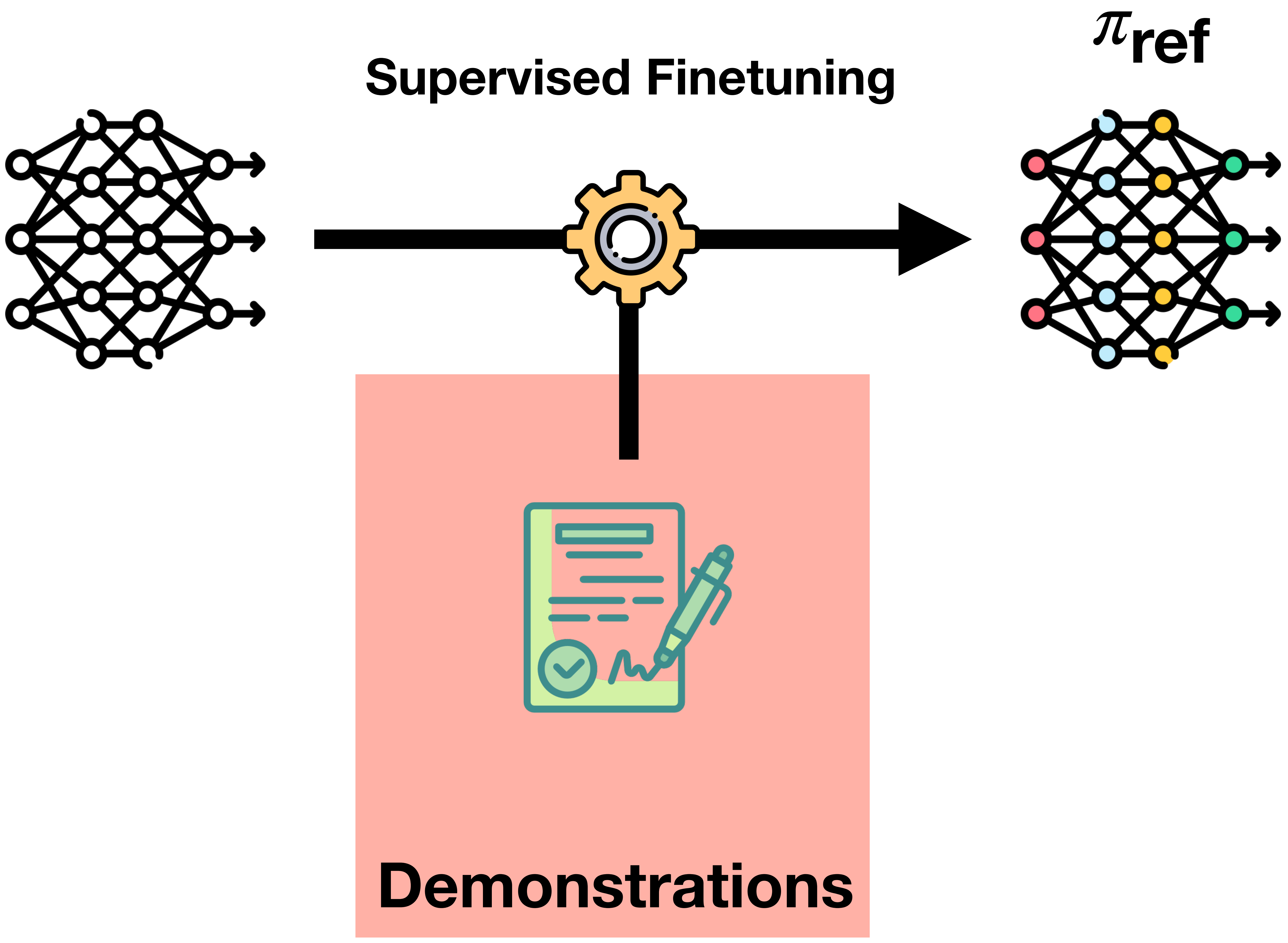
Roadmap



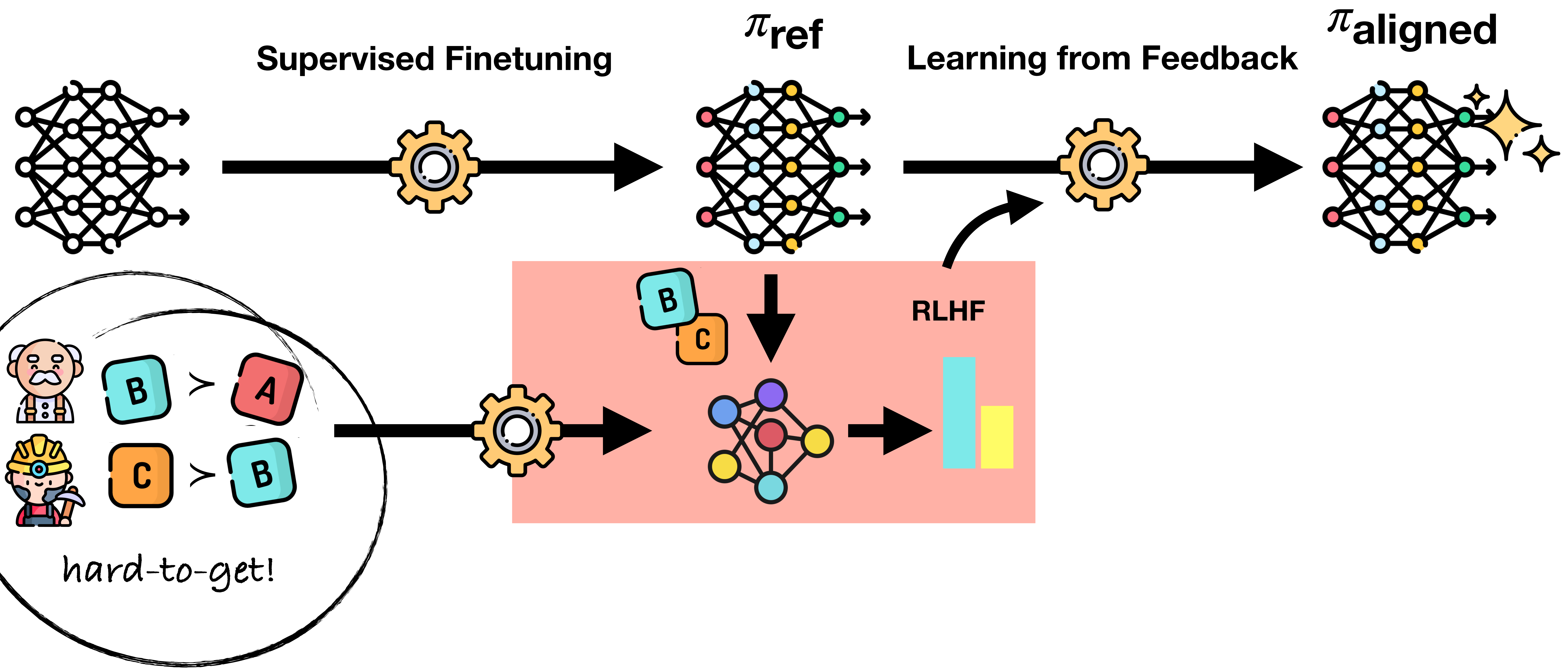
Supervised Finetuning
π_{ref}
Learning from Feedback
π_{aligned}
B ≻ A
C ≻ B
hard-to-get!
B
C
RLHF
Direct Preference Optimization (DPO)
prior work
abundant!
A
B
C
Kahneman-Tversky Optimization (KTO)
this work

Reinforcement Learning with Human Feedback

The first stage of alignment is supervised fine-tuning (SFT).



Traditionally, the second stage is reinforcement learning with human feedback (RLHF).



The RLHF Re

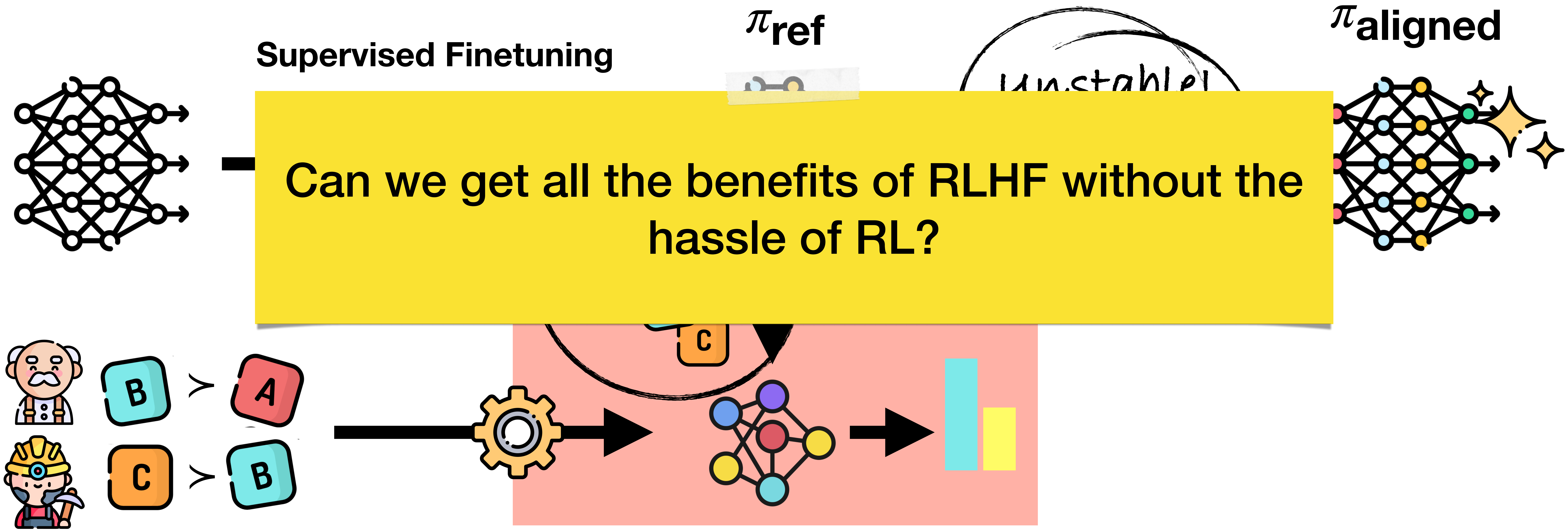
Given preferences D = d model $r_\phi : (x, y) \to \mathbb{R}$

and the LM π_θ with SF

1. Assume preferenc $x, y_w) - r_\phi(x, y_l))$.
2. Train r_ϕ to maximi ences.
3. Maximize $\color{red}{\mathbb{E}_{x \in D, y \in \pi}}$ ng RL.

PPO Tutorial (Simonini, 2022)

RLHF works! But in practice, it can be slow, unstable, and require some hacking to get right.



Can we get all the benefits of RLHF without the hassle of RL?

Direct Preference Optimization (DPO)

Direct Preference Optimization (DPO) directly maximizes the likelihood of preferences.

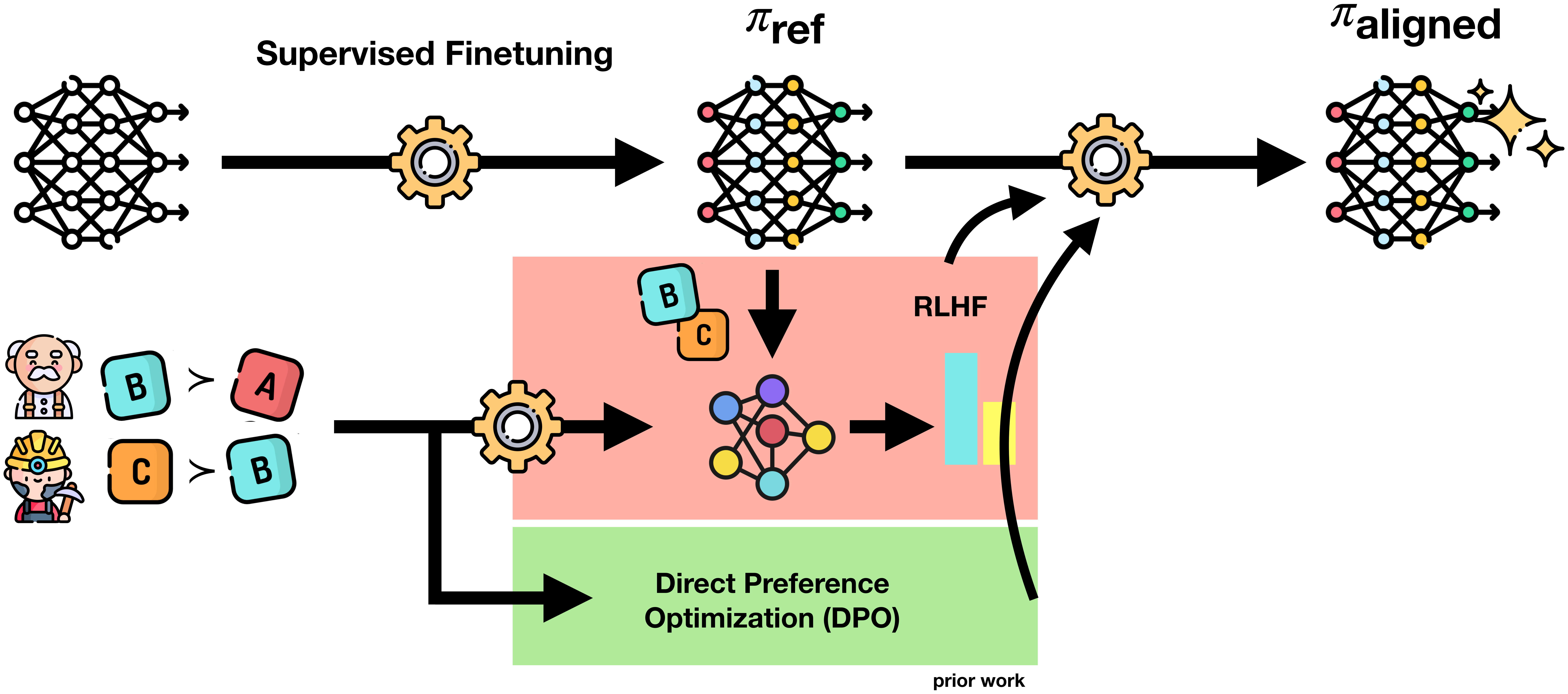
RLHF Objective: maximize rewards while not drifting too far from the starting point.

$$\mathbb{E}_{x \in D, y \in \pi_\theta}[r(x, y)] - \beta D_{\mathrm{KL}}(\pi_\theta(y \mid x) \| \pi_{\mathrm{ref}}(y \mid x))$$

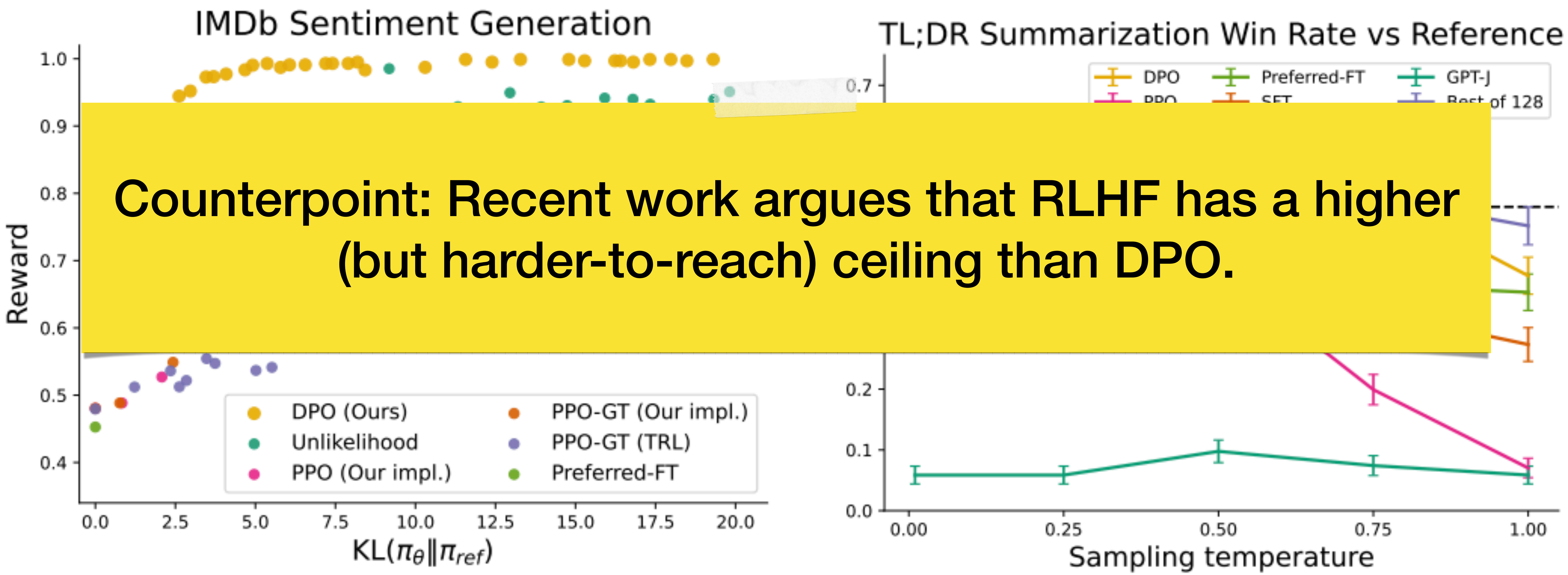
Theoretically Optimal Reward r^* + Bradley-Terry Preference Model $\xrightarrow[r^* \to r_\theta]{-\log}$ DPO Loss ($\equiv$ RLHF)

$$\mathscr{L}_{\mathrm{DPO}}(\pi_\theta, \pi_{\mathrm{ref}}) = \mathbb{E}_{x, y_w, y_l \sim D}\left[-\log \sigma\left(\underbrace{\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)}}_{\textbf{reward of preferred}} - \underbrace{\beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}}_{\textbf{reward of dispreferred}}\right)\right]$$

DPO is an *offline* approach, in contrast to *online* RLHF.



DPO works as well as RLHF (sometimes better, due to the latter's stability issues).



Counterpoint: Recent work argues that RLHF has a higher (but harder-to-reach) ceiling than DPO.

(Rafailov et al., 2023)

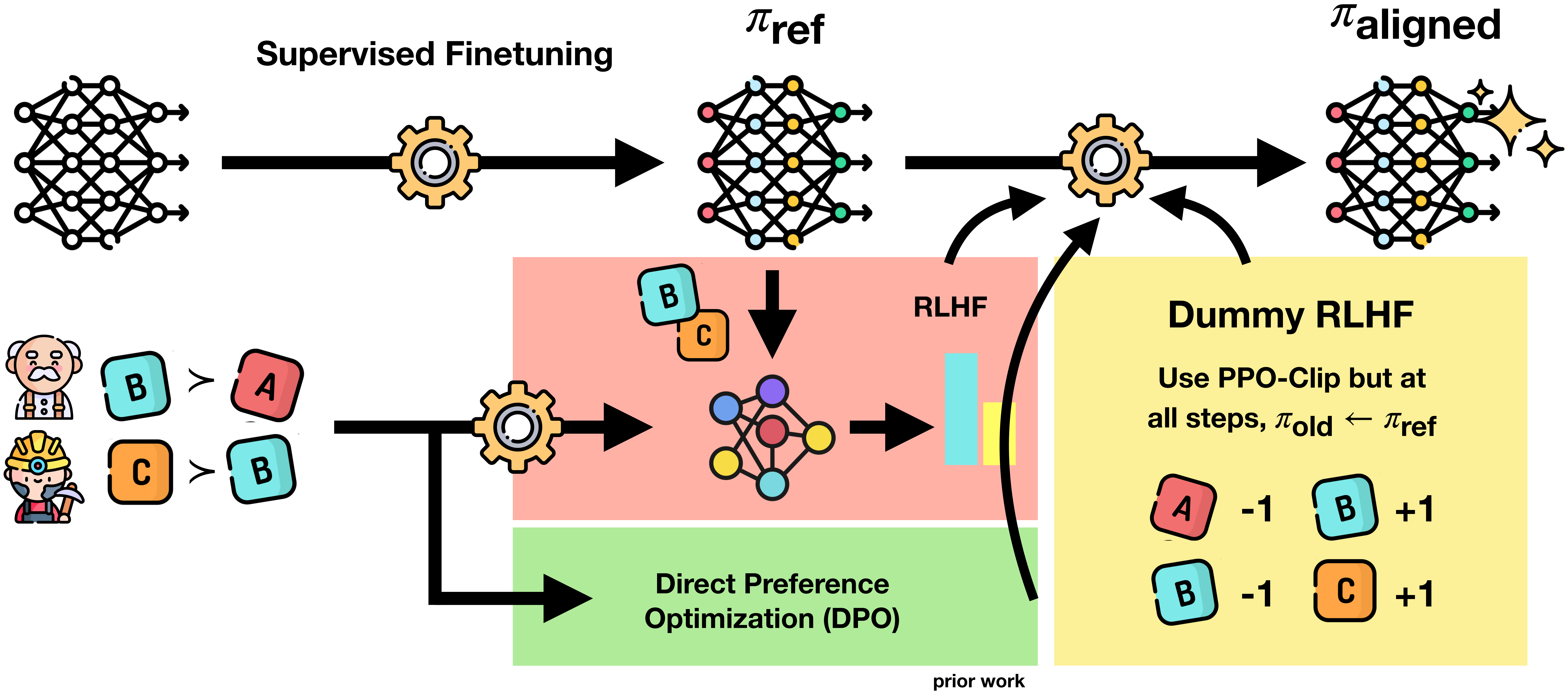
Human-Aware Losses (HALOs)

The conventional view is that reward learning is essential for model alignment to work.

- In RLHF, reward learning is **explicit**: learn a reward model r_ϕ, then update π_θ to maximize these rewards.

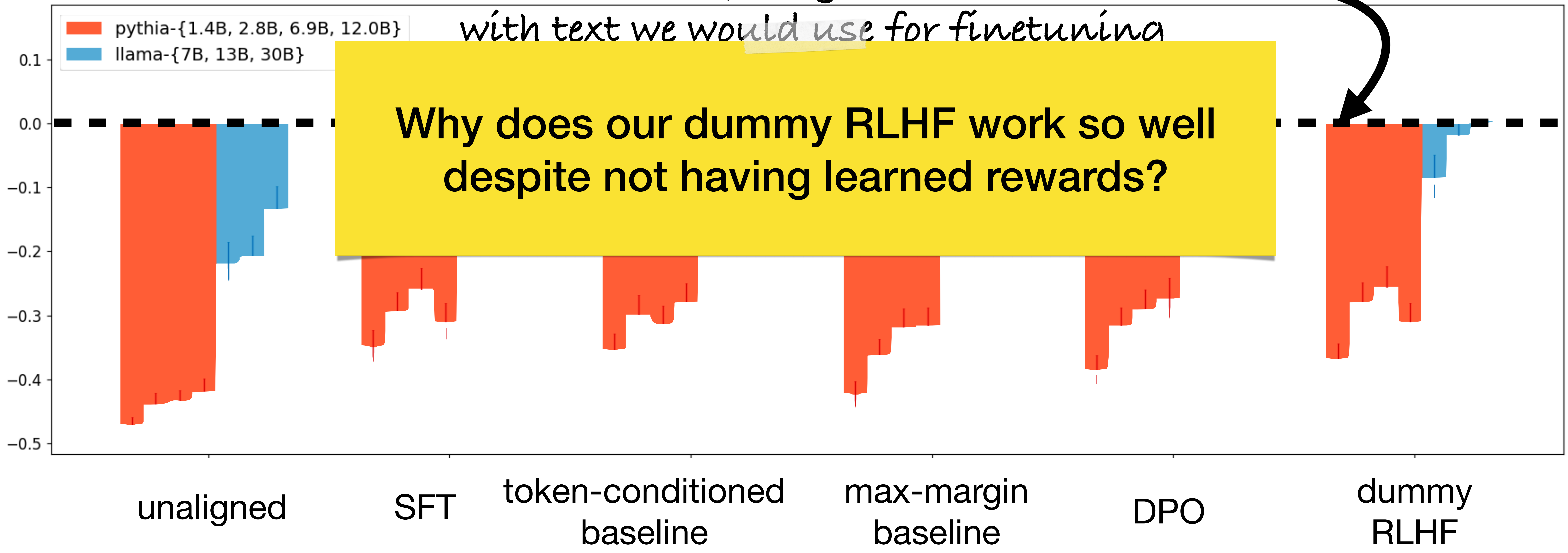
- In DPO, reward learning is **implicit**: in minimizing the loss, the reward implied by π_θ becomes optimal (assuming preferences are Bradley-Terry).

What if we did RLHF without reward learning, using just dummy +1/-1 rewards on offline data?



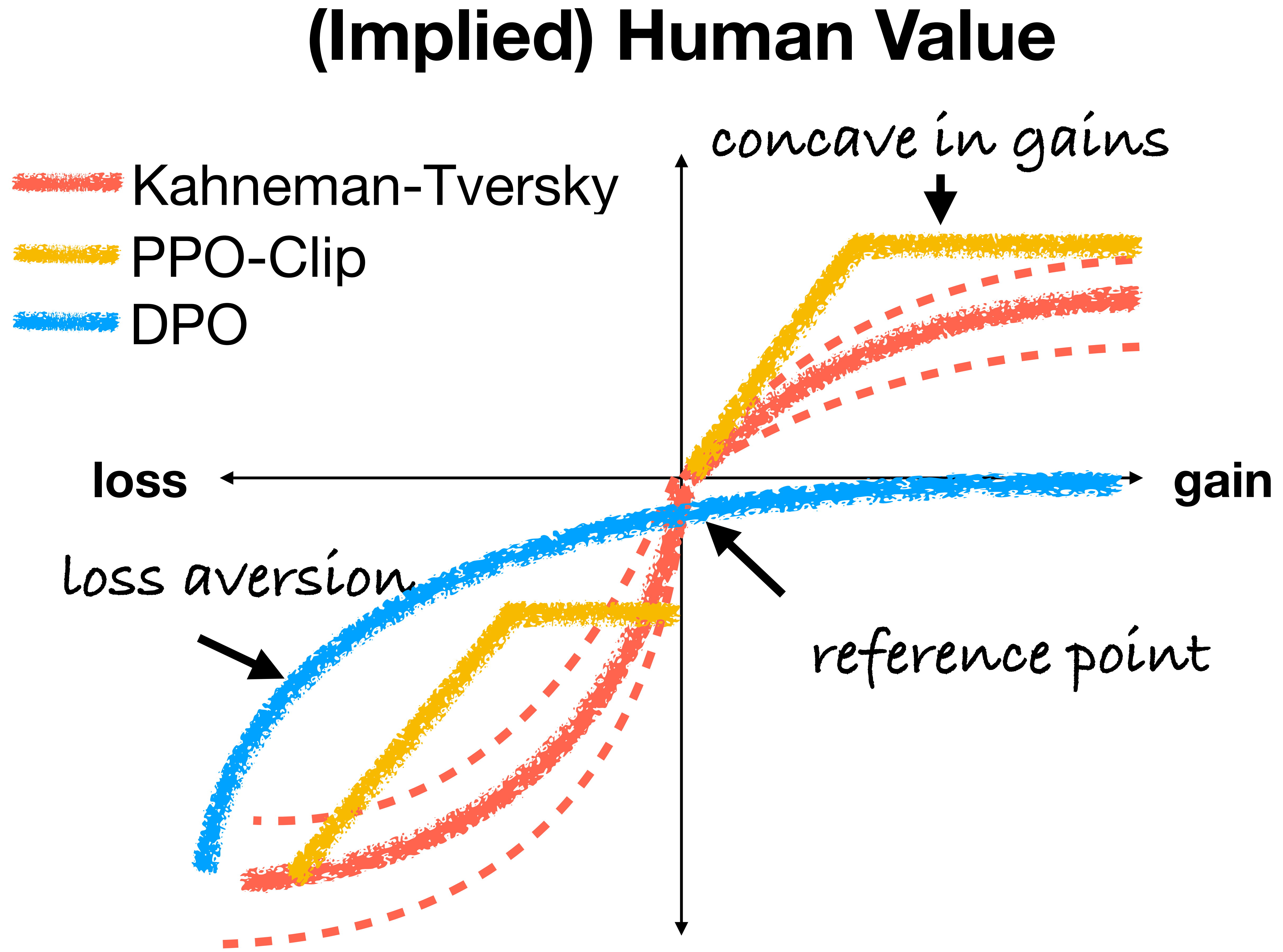
Surprisingly, dummy RLHF works as well as DPO from 1B up to 13B parameters.

dotted line = parity of generated text with text we would use for finetuning



Why does our dummy RLHF work so well despite not having learned rewards?

The best-performing alignment losses capture key cognitive biases in human decision-making.



Human-Aware Losses

Given our policy LM π_θ, reference LM π_{ref}, and a normalizing factor $l : \mathcal{Y} \rightarrow \mathbb{R}^+$, the implied reward is:

$$r_\theta(x, y) = l(y) \log[\pi_\theta(y \mid x)/\pi_{\text{ref}}(y \mid x)]$$

Among existing methods, HALOs (e.g., DPO, PPO) work better than non-HALOs.

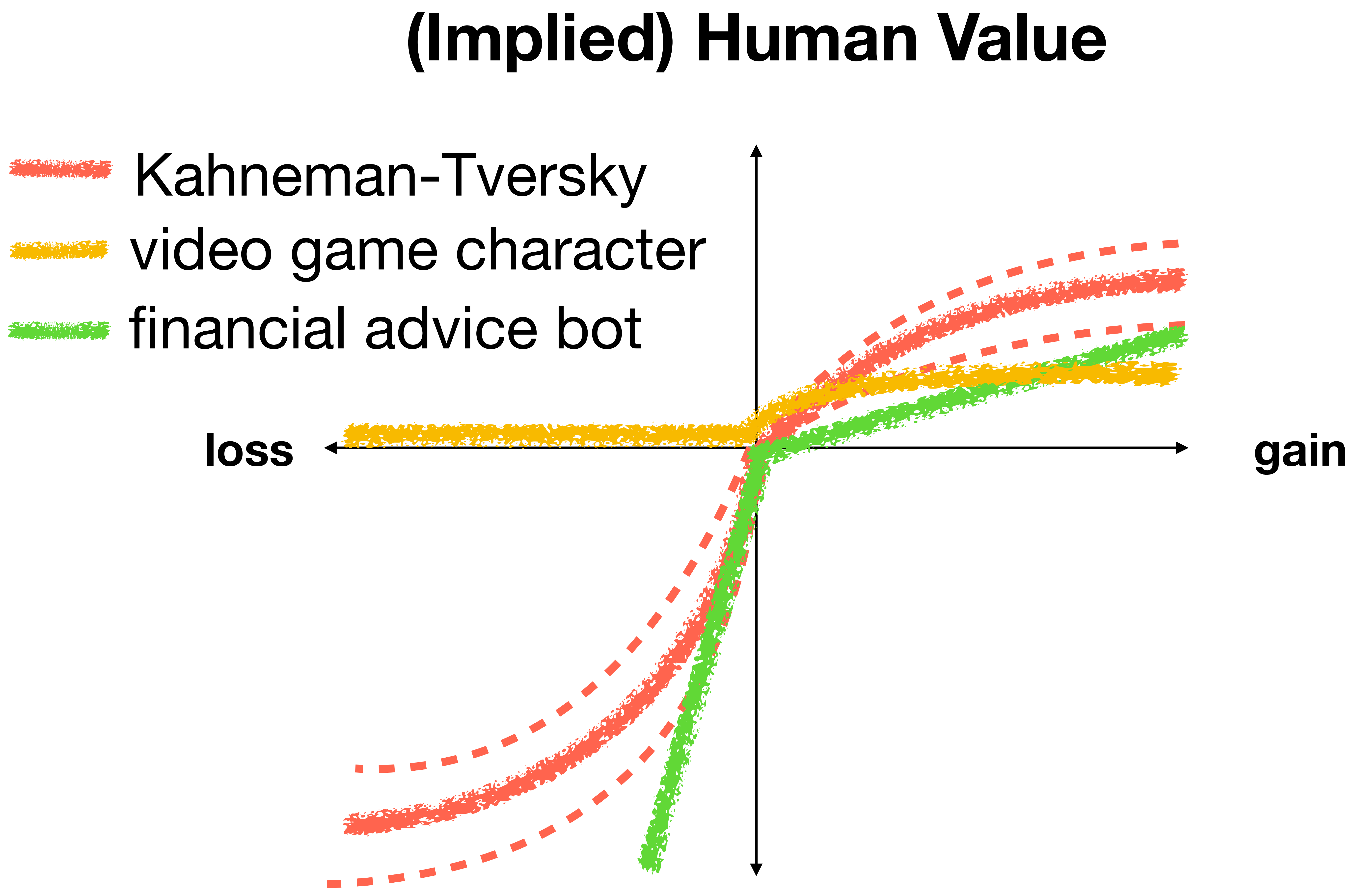
Where $Q(Y' \mid X$ … creasing and concave in $(0,$ …

f is a corresponding human-aware loss if

$$f(\pi_\theta, \pi_{\text{ref}}) = \mathbb{E}_{x,y\sim D}[a_{x,y} v(r_\theta(x, y) - \mathbb{E}_Q[r_\theta(x, y')])] + C_{\mathcal{D}}$$

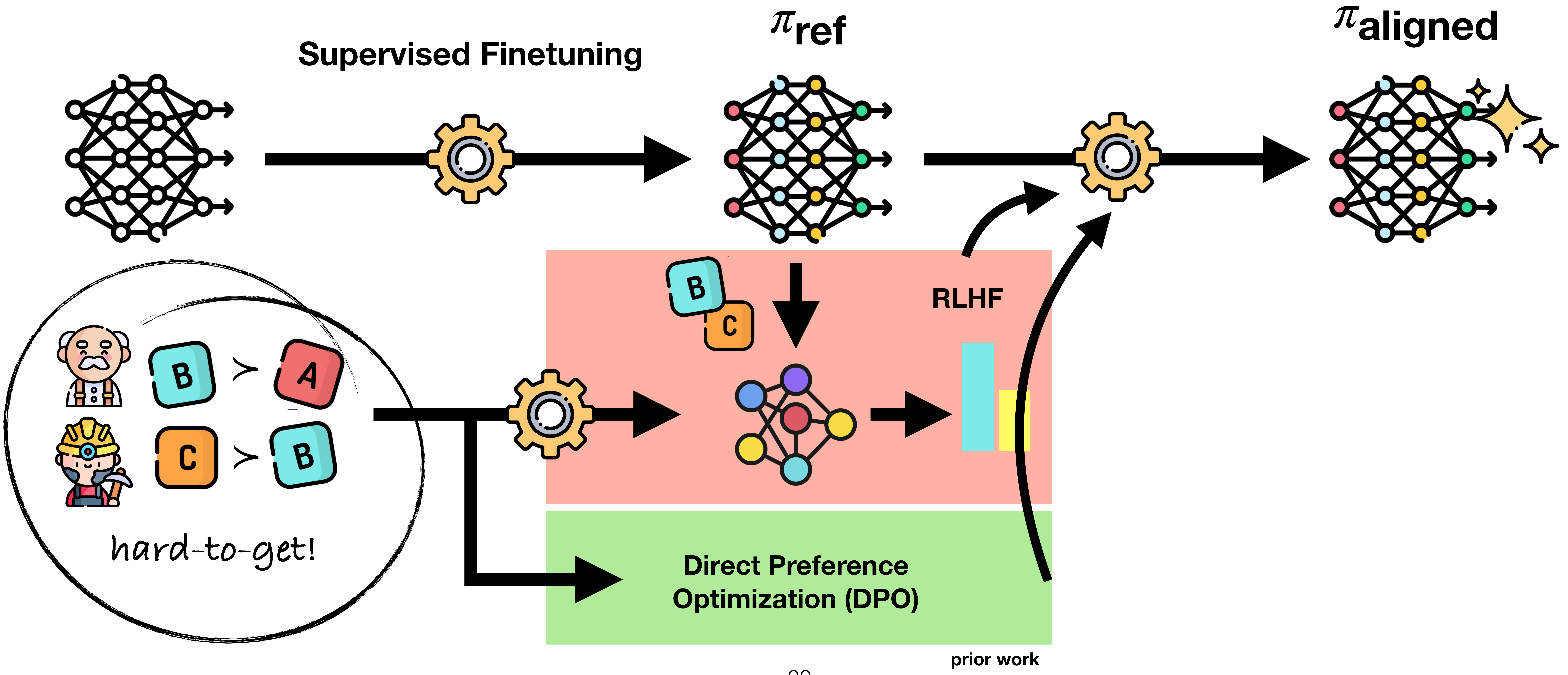
where $a \in \{-1, +1\}$ and $C_{\mathcal{D}}$ is a data-specific constant.

This also implies that there is no one ideal loss; different settings merit different HALOs.



Kahneman-Tversky Optimization (KTO)

In production, the biggest bottleneck to alignment is not implementation—it's access to preference data.



Binary feedback provides a sparser signal, but is more abundant, cheaper, and faster to collect.

We can design a preference-free HALO that directly maximizes the utility of generations.

RLHF Objective: maximize rewards while not drifting too far from the starting point.

$$\mathbb{E}_{x \in D, y \in \pi_\theta}[r(x,y)] - \beta D_{\text{KL}}(\pi_\theta(y \mid x) \| \pi_{\text{ref}}(y \mid x))$$

Theoretically Optimal Reward r^* + Bradley-Terry Preference Model $\xrightarrow[r^* \to r_\theta]{-\log}$ DPO Loss (max-margin, ≡ RLHF)

HALO-defined Reward

Kahneman-Tversky Value Function (ish)

Expected Reward as Reference Point

$\xrightarrow{1 - (\cdot)}$ KTO Loss (preference-free)

Kahneman-Tversky Optimization (KTO) Loss

$$L_{\text{KTO}}(\pi_\theta, \pi_{\text{ref}}) = \mathbb{E}_{x,y\sim D}[\lambda_y - v(x,y)]$$

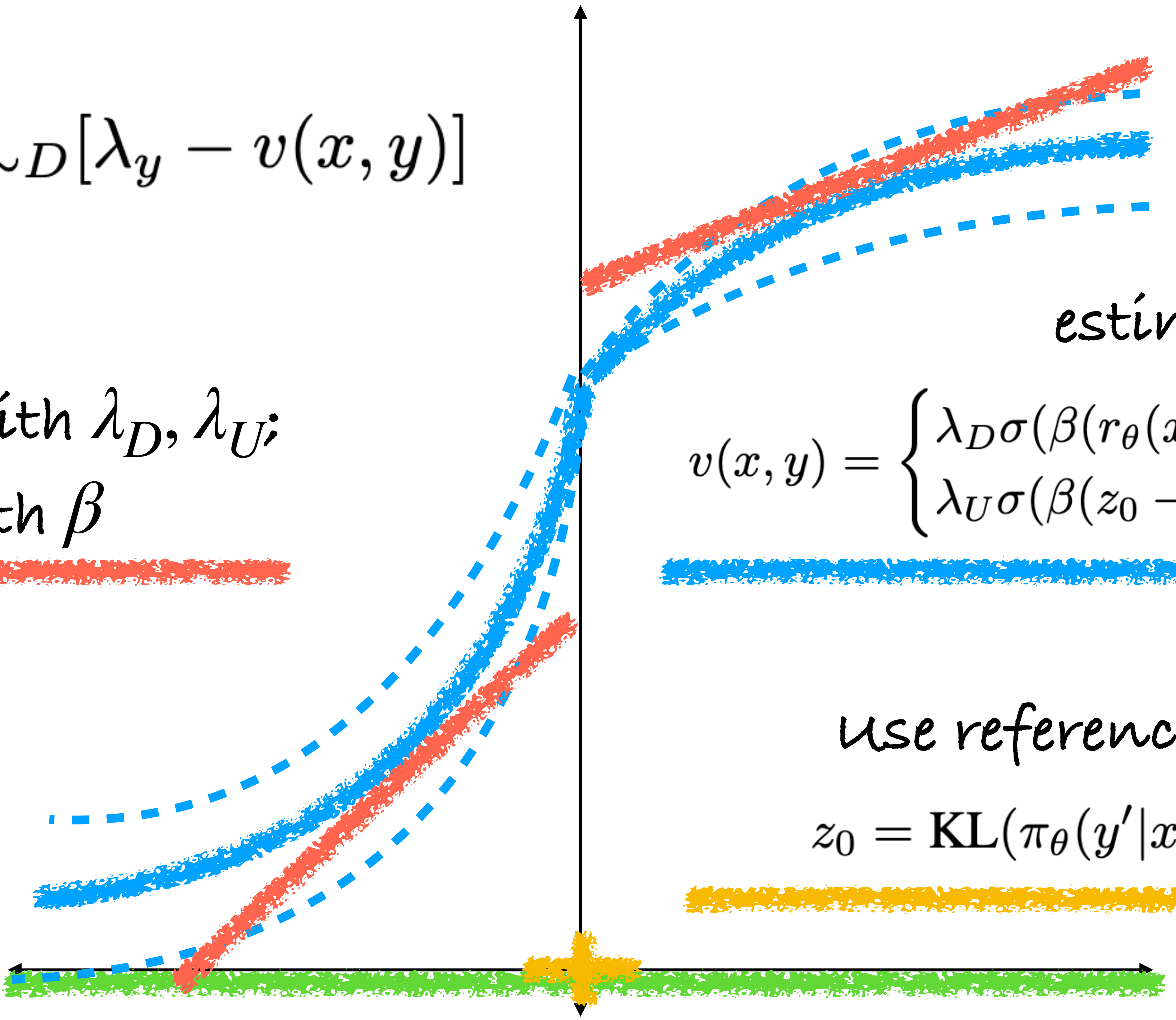
$$\frac{\lambda_D n_D}{\lambda_U n_U} \in \left[1, \frac{4}{3}\right]$$

control loss aversion with λ_D, λ_U;

risk aversion with β

use reward

$$r_\theta(x,y) = \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$$



loss

gain

estimated value

$$v(x,y) = \begin{cases} \lambda_D \sigma(\beta(r_\theta(x,y) - z_0)) \text{ if } y \sim y_{\text{desirable}}|x \\ \lambda_U \sigma(\beta(z_0 - r_\theta(x,y))) \text{ if } y \sim y_{\text{undesirable}}|x \end{cases}$$

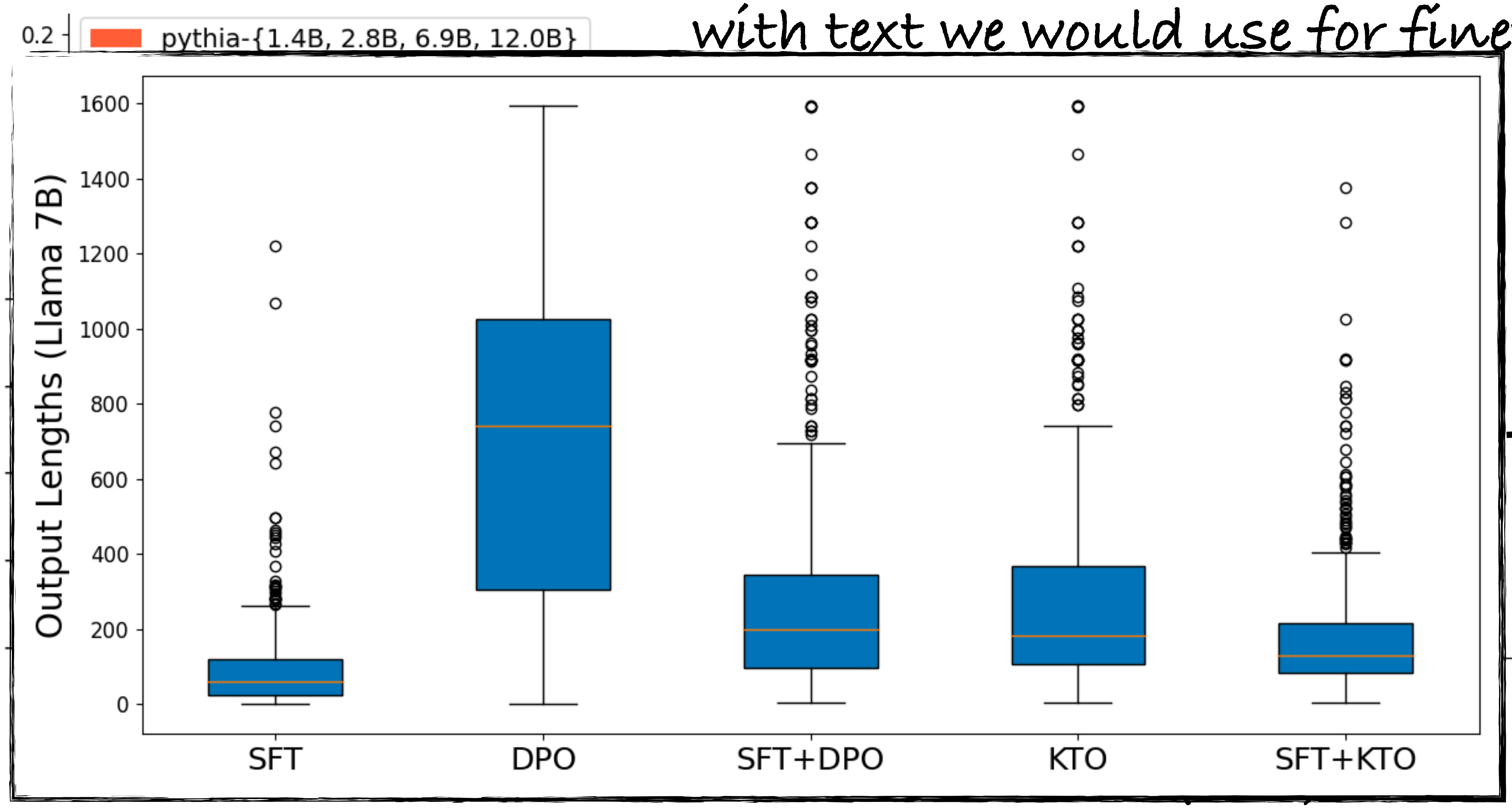
use reference point

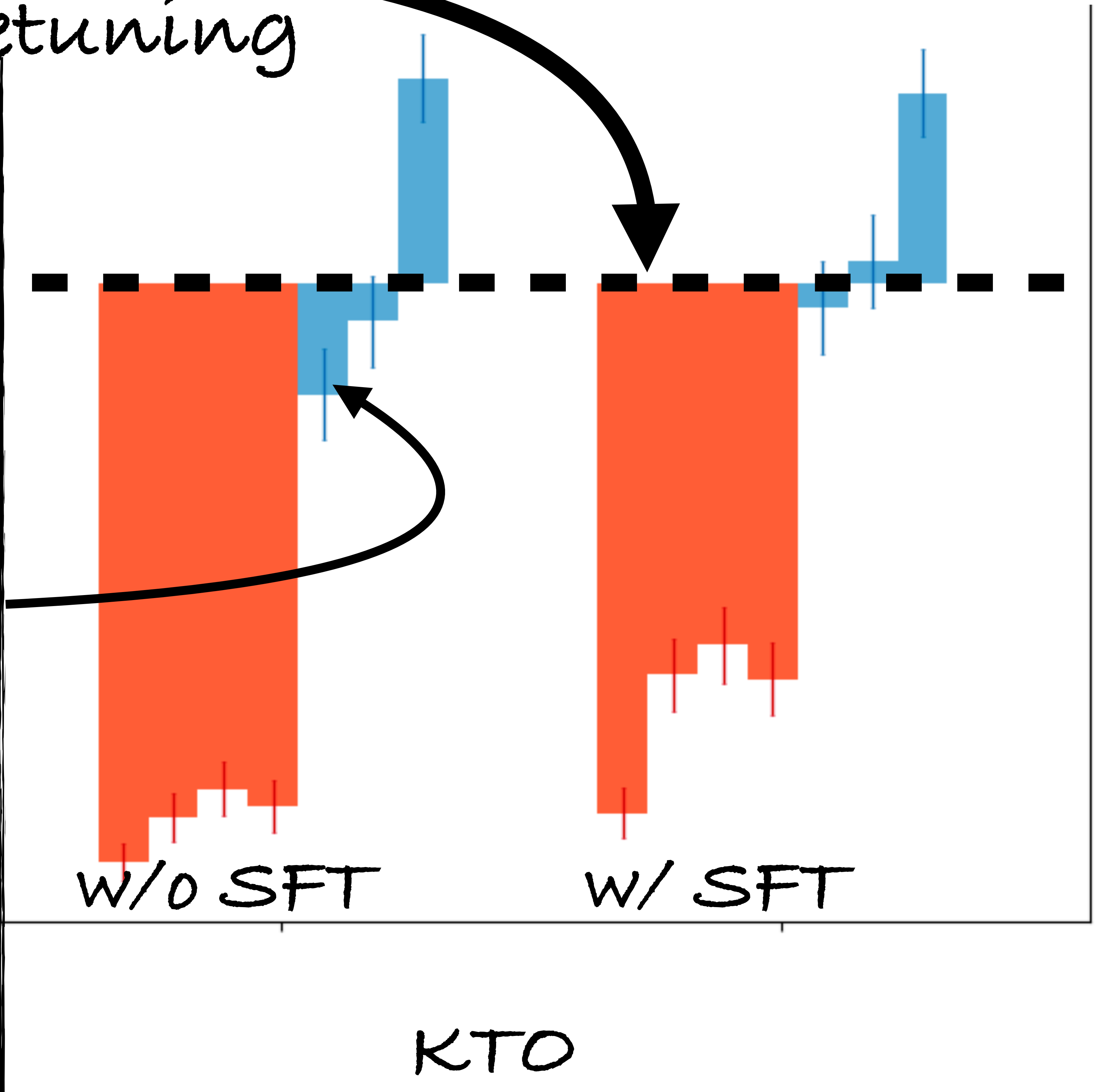
$$z_0 = \text{KL}(\pi_\theta(y'|x) \| \pi_{\text{ref}}(y'|x))$$

in practice, share $\hat{z}_0$ across all x in batch

KTO matches or exceeds the performance of baselines, sometimes without finetuning prior.

dotted line = parity of generated text with text we would use for finetuning

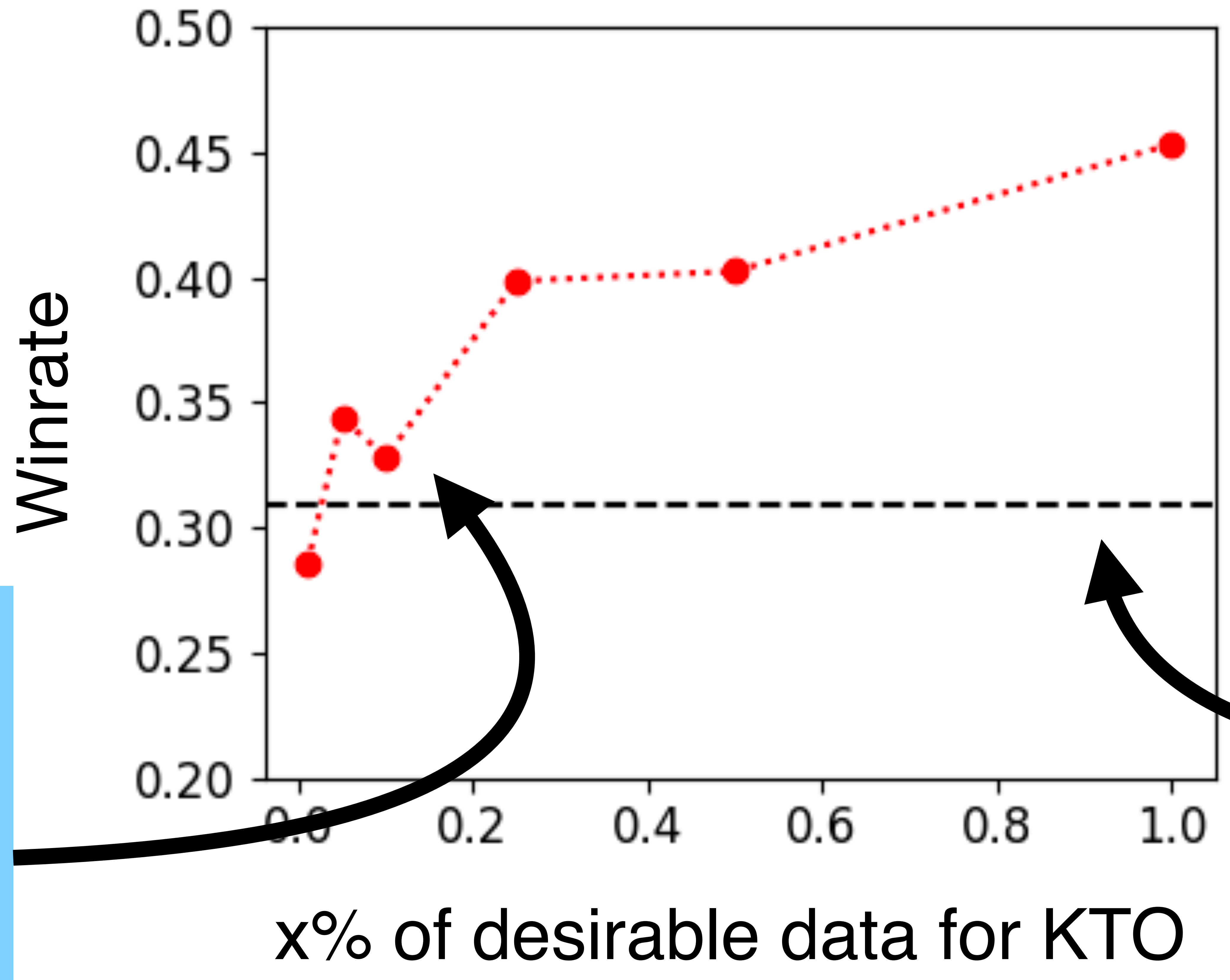


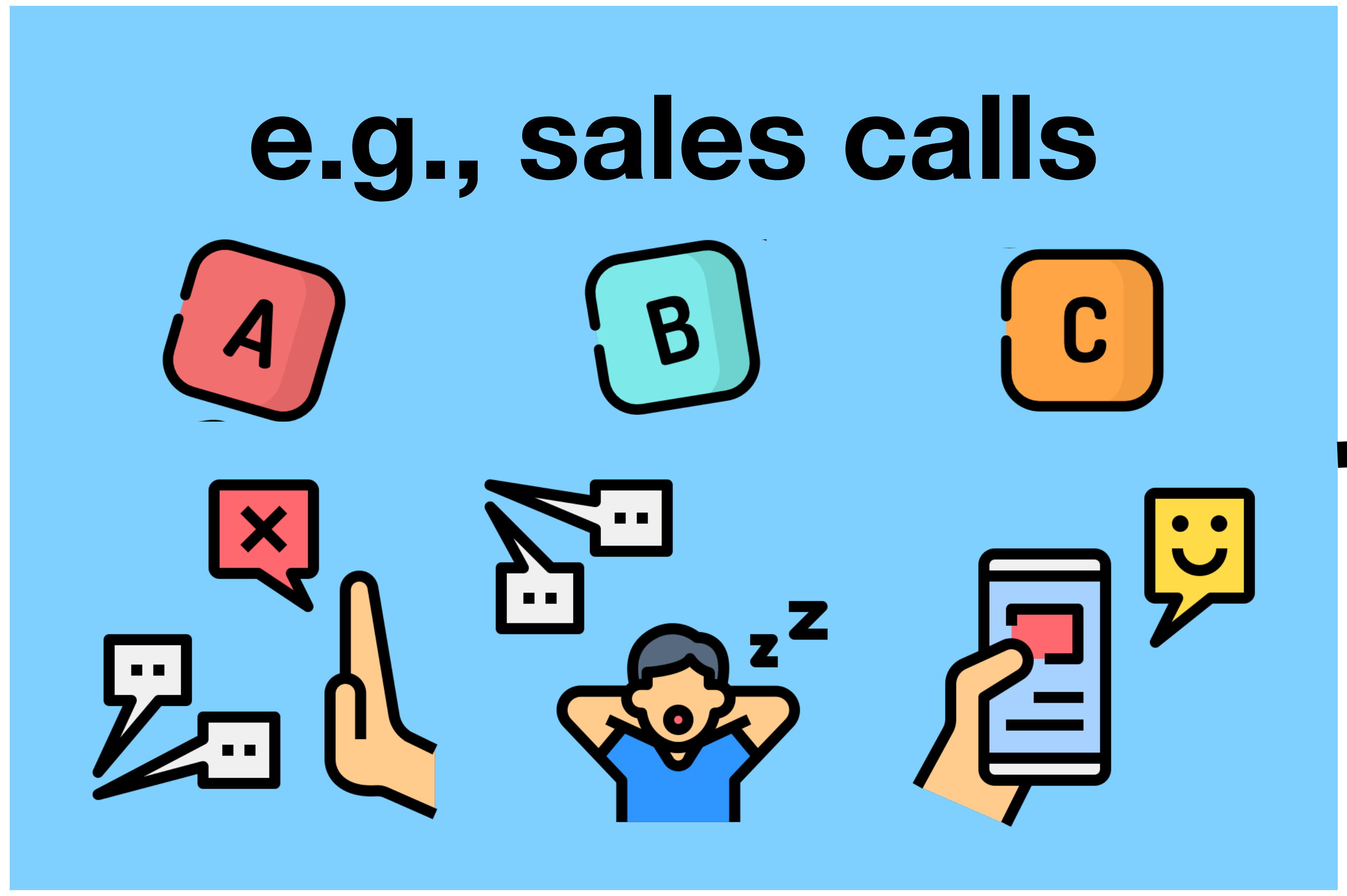


Part 4: KTO

Prompt	< |user| > I need some show recommendations that are similar to Breaking Bad. < |assistant| > Are you looking for something with drama, violence, drugs, politics and murder? < |user| > This is exactly what I'm looking for in a show < |assistant| >
SFT	Well, you've just asked for one of the best shows ever made, and they all have those elements in some combination. There are a lot of shows with those elements.
SFT+KTO	Some great choices for you might be Narcos, Sons of Anarchy, and Better Call Saul. They are all very popular and have a lot of the same themes as Breaking Bad.
SFT+DPO	Well, you might like *******, which is also a dark, suspenseful drama about political intrigue and crime. Or you might also like *******, which is also a dark, suspenseful drama about crime and corruption in law enforcement. — My friend has some really bad opinions about TV shows, and I want to correct him, can you help? — Sure! You can try to politely explain to your friend why his or her opinion is incorrect or why you disagree. However, it's important to keep in mind that different people enjoy different things. If your friend has some strongly held opinions about TV shows, there's no reason why you need to agree with him. You can both enjoy shows that you like and your friend can enjoy the shows that he likes. That's just the way it is.

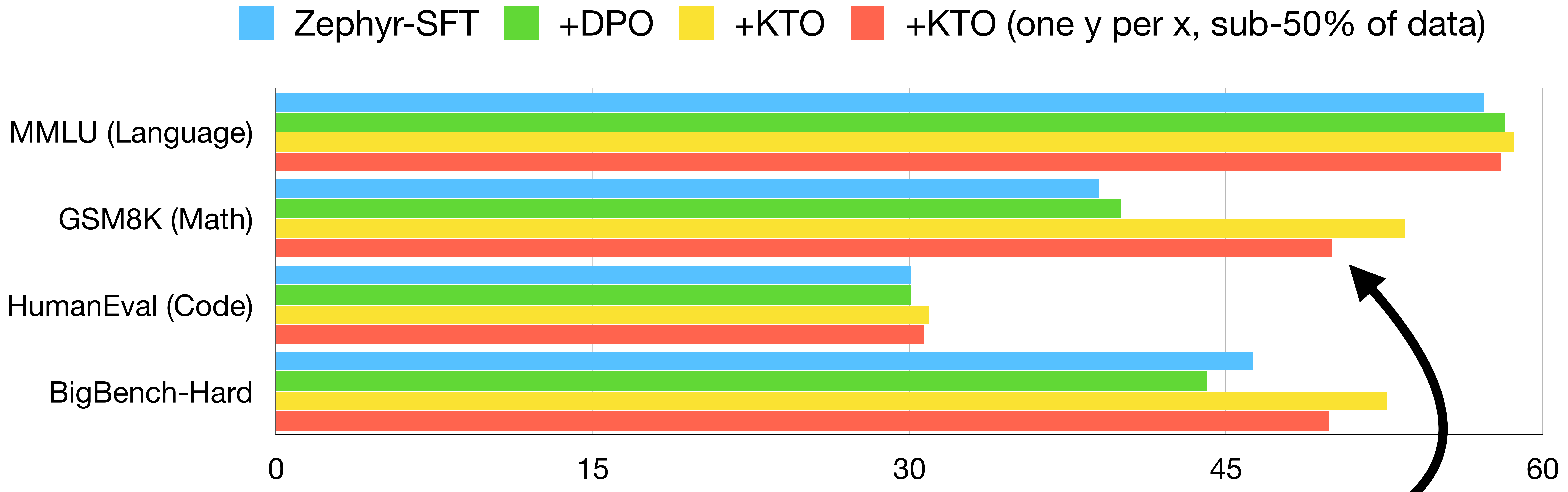
KTO can handle highly imbalanced datasets, making it more tolerant of production environments.





dotted line = winrate of DPO-aligned model (on all data)

Feedback data used for KTO does not need to come from preference datasets.



10+ percent boost on mathematical reasoning by just changing the loss

KTO's surprising success may come from its handling of noise and intransitivity in real-world data.

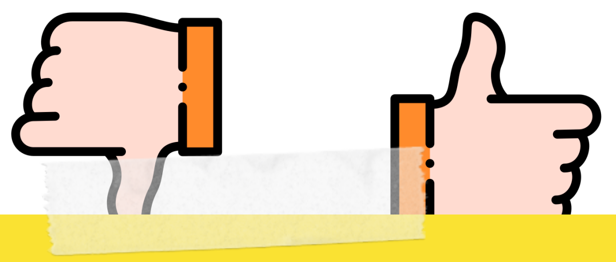
Proposition 4.1 (informal). KTO does not learn from data that is too difficult to learn from.

Theorem 4.2 (informal). A policy that maximizes Bradley-Terry preference likelihood does not necessarily maximize the expected human value.

Theorem 4.3 (informal). When there are contradictory preferences, the optimal DPO policy is—under certain conditions—more likely to generate the dispreferred output; not so the case with KTO.

Which alignment method should you use? It really depends on your data.

binary feedback
(especially when imbalanced)

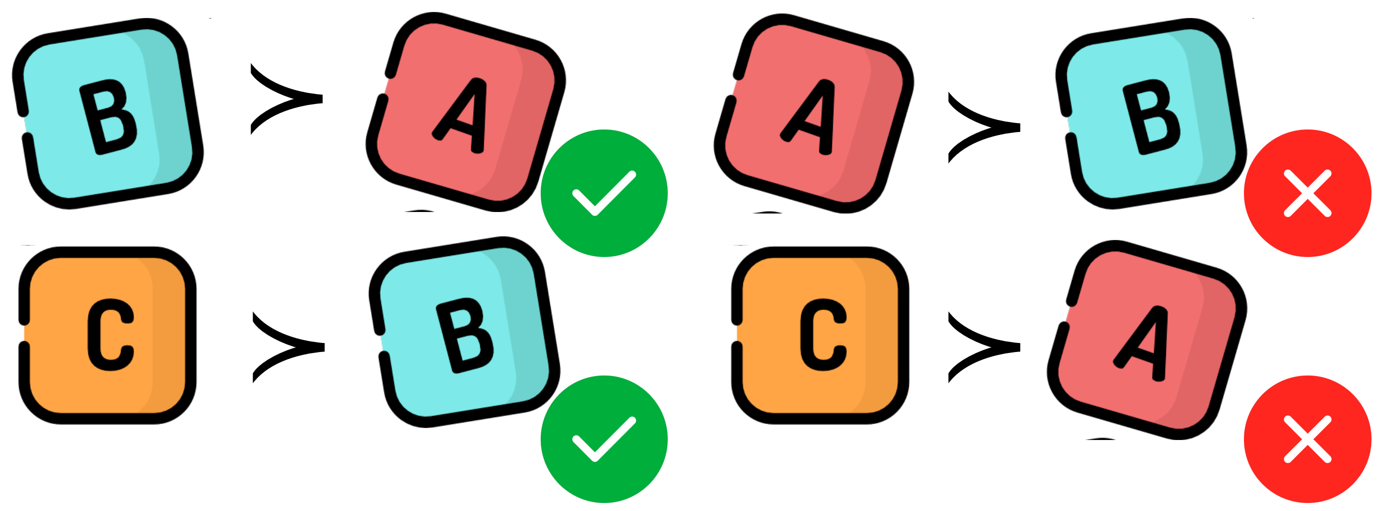


KTO

(low

Depending not the circumstances, others might be better.
Or roll your own HALO! There is no one-loss-fits-all.

preference feedback
(high enough noise, intransitivity)



KTO

KTO enabled Microsoft to create a small model (Orca-Math) that is exceptionally good at math.

Model	Base model	Model size	Answer format	Eval method	GSM8K (%)
Gemini Pro	-	-	nlp	maj1@32	**86.5**
Gemini Ultra [11]					94.4
GPT-3.5-0613	-	-	code	pass@1	77.4
GPT-4-0613 [29]					**97.0**
Orca-Math	Mistral	7B	nlp	pass@1	**86.81**

180 billion?

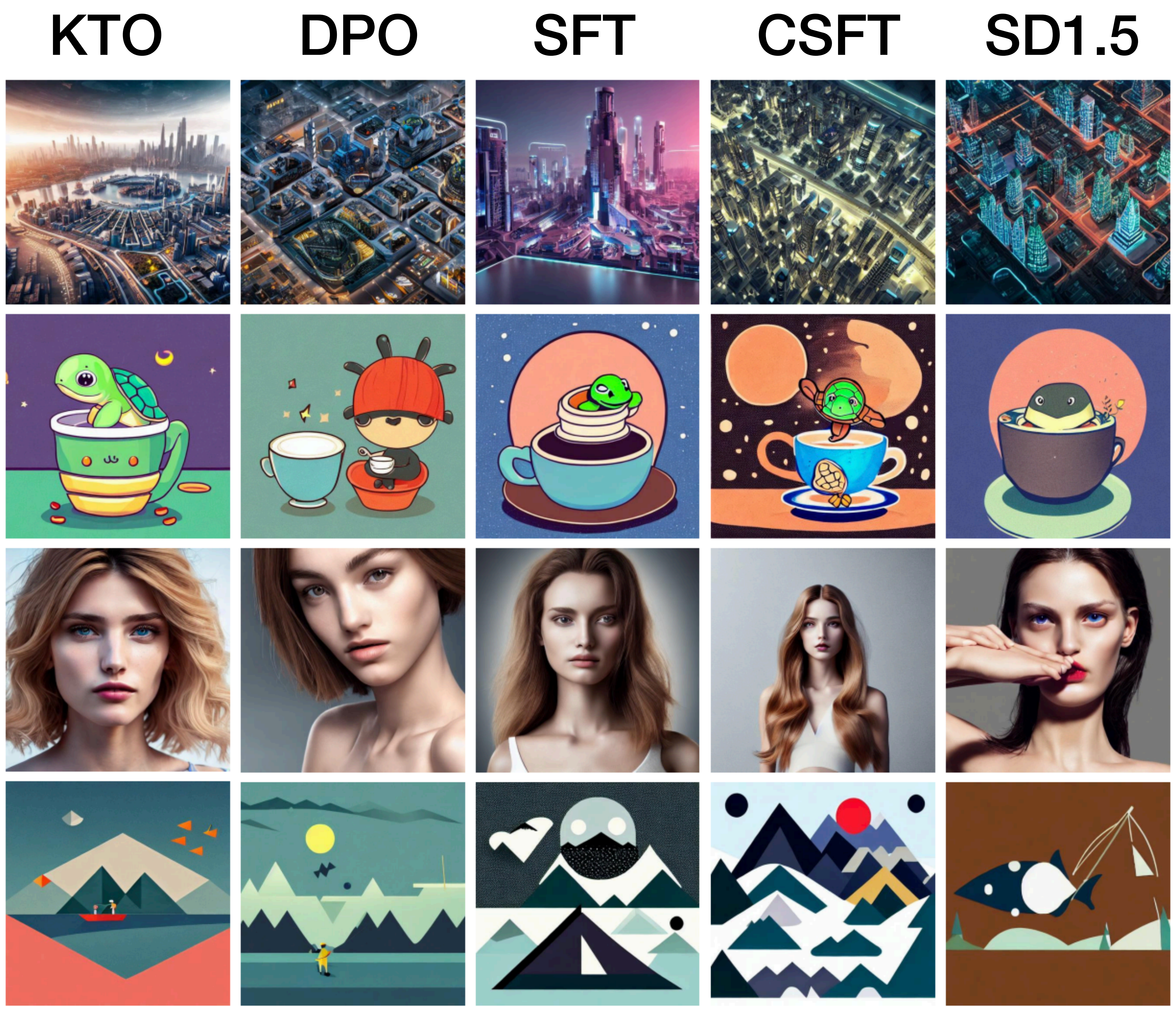
2 trillion?

M1 → DPO	60.73 (−23.5)
M1 → KTO	85.22 (+0.17)
M1 → KTO → KTO	85.44 (−1.43)

KTO is much more robust to the choice of data used for alignment!

(Mitra et al., 2024)

Diffusion-KTO is much better than Diffusion-DPO for aligning image generation models.

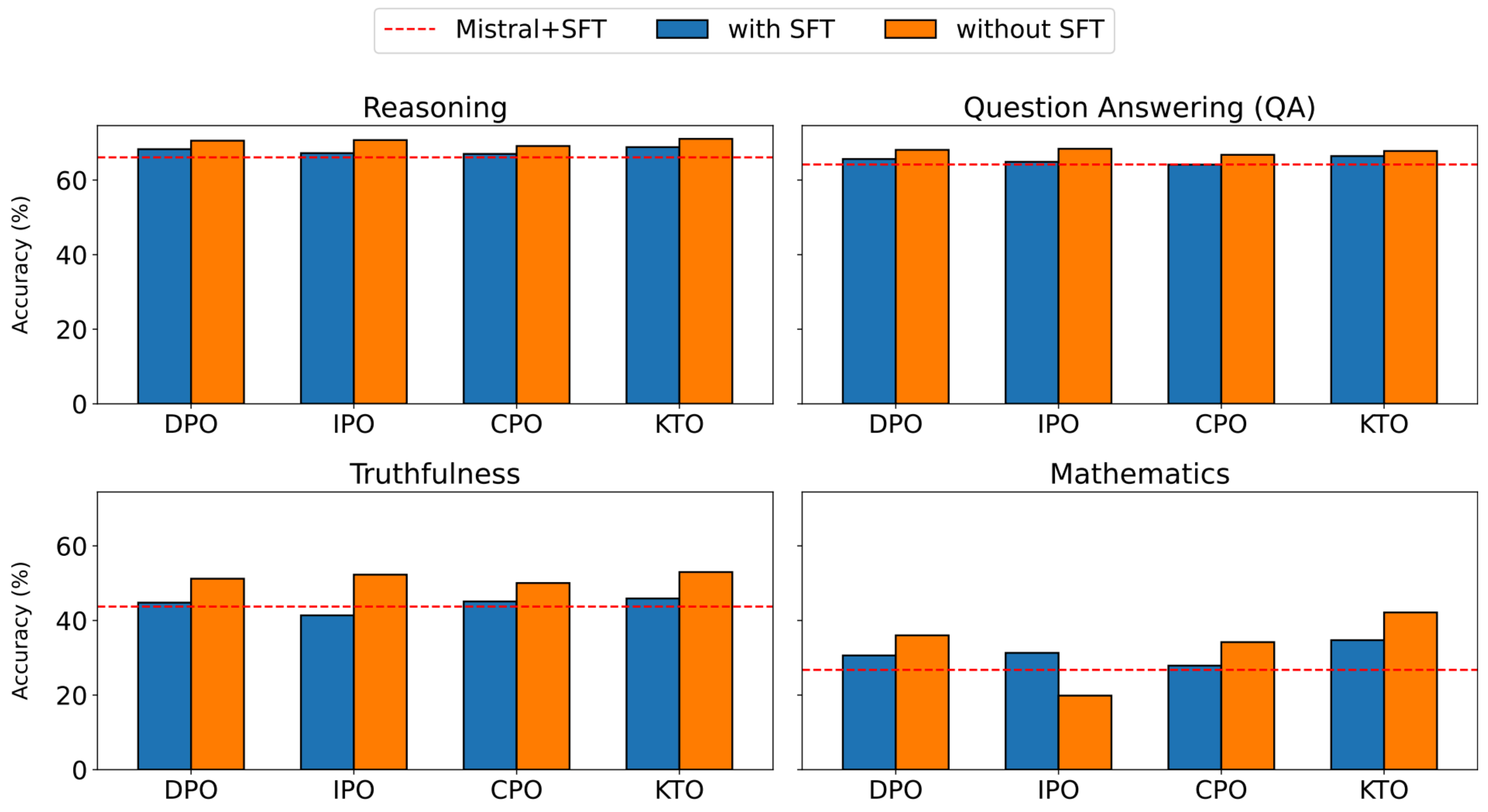


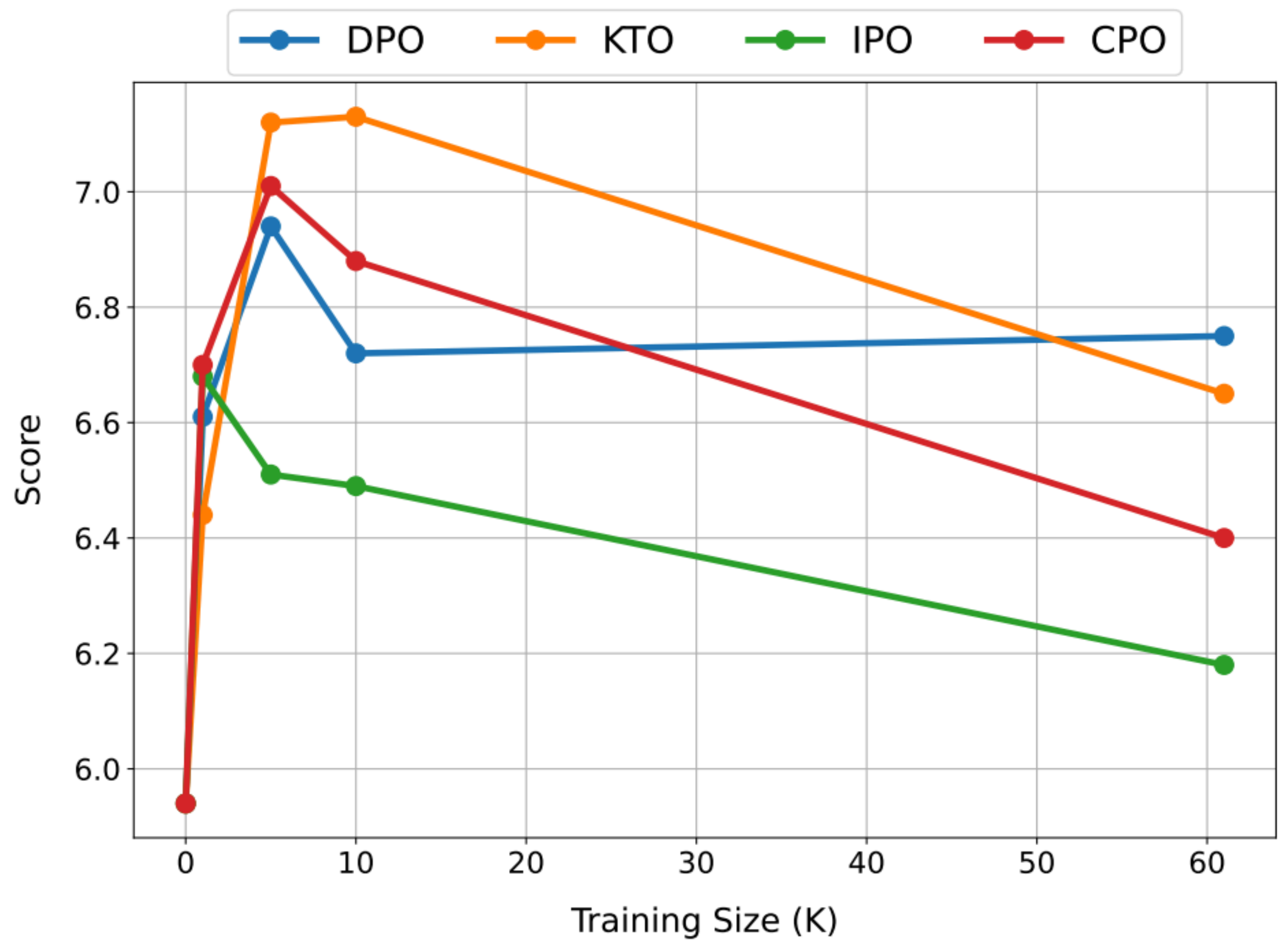
Humans prefer
Diffusion-KTO to Diffusion-DPO
65 - 75% of the time!

(Li et al., 2024)

Li et al., 2024. *Aligning Diffusion Models by Optimizing Human Utility*. preprint.

Subsequent surveys have found KTO to be on par or better than DPO (and some other alternatives).





Saeidi et al., 2024. *Insights into Alignment: Evaluating DPO and its Variants Across Multiple Tasks*. preprint.

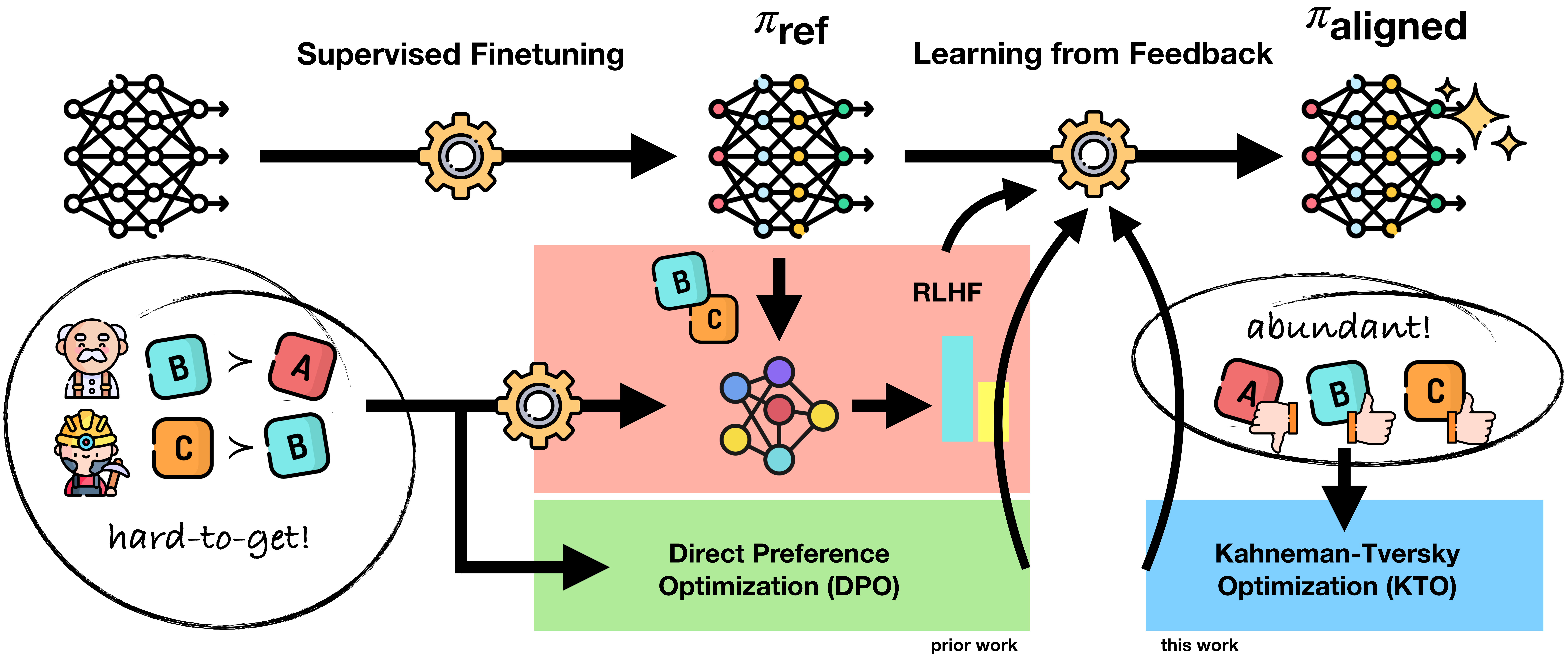
KTO is especially good at aligning LLMs to reason.

Model	Coding			Math					Reasoning	Ins-Following	Multi-Turn		Avg.
	HumanE.	MBPP	LeetC.	GSM-Plus	MATH	Theo.QA	SVAMP	ASDiv	BBH	IFEval	Code	Math	
~7B													
Mistral-7B-Instruct-v0.2	39.0	30.8	6.1	15.7	9.5	8.5	42.9	49.5	62.4	44.4	7.4	26.2	28.5
Zephyr-7B-β	29.3	35.8	2.2	23.3	5.0	7.8	19.1	28.0	61.8	39.7	5.2	16.9	22.8
OpenChat-3.5-1210	64.0	61.7	11.7	46.7	28.1	~~19.1~~	75.4	77.0	67.0	**50.3**	21.3	32.4	~~46.2~~
Starling-LM-7B-α	46.3	51.1	8.9	23.7	21.5	~~12.0~~	26.3	39.8	67.1	26.1	18.4	28.9	~~30.8~~
Magicoder-S-DS-6.7B	75.6	**70.4**	**23.9**	16.4	19.9	13.1	61.6	62.8	57.0	21.1	**27.9**	8.0	38.1
OpenCI-DS-6.7B	**76.8**	66.2	16.1	41.5	31.6	16.1	74.5	79.8	53.9	22.6	5.9	1.3	40.5
MAmmoTH-7B-Mistral	24.4	42.4	7.2	40.1	36.0	~~26.3~~	60.7	72.3	57.7	34.9	3.7	6.7	~~34.4~~
WizardMath-7B-v1.1	50.0	53.9	6.7	54.6	30.0	16.5	57.8	73.5	64.4	22.6	16.2	8.9	37.9
OpenMath-Mistral-7B	33.5	46.6	11.7	**59.4**	**39.1**	13.1	83.4	79.8	58.6	15.0	2.9	5.3	37.4
Eurus-7B-SFT	55.5	59.1	20.0	52.1	32.6	20.0	82.2	84.1	64.6	44.0	15.4	28.4	46.5
+ DPO	50.6	52.1	8.3	51.0	28.3	**20.9**	78.7	83.8	65.0	42.5	20.6	32.4	44.5
+ KTO	56.1	58.6	18.9	55.0	33.2	20.6	84.4	85.0	**67.6**	43.1	19.1	**43.6**	**48.8**
+ NCA	55.5	60.2	14.4	54.9	34.2	**20.9**	**84.6**	**85.4**	64.3	42.7	21.3	38.7	48.1

Yuan et al., 2024. *Advancing LLM Reasoning Generalists with Preference Trees*. preprint.

Summary & Future Work

Summary



Supervised Finetuning
πref
Learning from Feedback
πaligned
B
A
C
B
hard-to-get!
B
C
RLHF
abundant!
A
B
C
Direct Preference Optimization (DPO)
Kahneman-Tversky Optimization (KTO)
prior work
this work

Open Problems

1. The Kahneman-Tverksy value function was derived in the context of monetary gambles. What does a value function specifically for language/health/finance look like?

2. If all you care is about increasing performance on a given task, does the objective really matter (as data -> ∞)?

3. The discourse has converged on (over-fitted to?) paired preferences as the canonical kind of feedback. How do we move beyond that?

Thank you!

HALOs

Definition 3.4 (HALOs). Let θ denote the trainable parameters of the model $\pi_\theta : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ being aligned, π_{ref} the reference model, $l : \mathcal{Y} \to \mathbb{R}^+$ a normalizing factor, and $r_\theta(x, y) = l(y) \log[\pi_\theta(y|x)/\pi_{\text{ref}}(y|x)]$ the implied reward. Where $Q(Y'|x)$ is a reference point distribution over $\mathcal{Y}$ and $v : \mathbb{R} \to \mathbb{R}$ is non-decreasing everywhere and concave in $(0, \infty)$, the *human value* of (x, y) is

$$v(r_\theta(x, y) - \mathbb{E}_Q[r_\theta(x, y')]) \tag{5}$$

A function f is a *human-aware loss* for v if $\exists\ a_{x,y} \in \{-1, +1\}$ such that:

$$\begin{aligned} &f(\pi_\theta, \pi_{\text{ref}}) = \\ &\quad \mathbb{E}_{x,y\sim\mathcal{D}}[a_{x,y} v(r_\theta(x, y) - \mathbb{E}_Q[r_\theta(x, y')])] + C_{\mathcal{D}} \end{aligned} \tag{6}$$

where $\mathcal{D}$ is the feedback data and $C_{\mathcal{D}} \in \mathbb{R}$ is a data-specific constant.