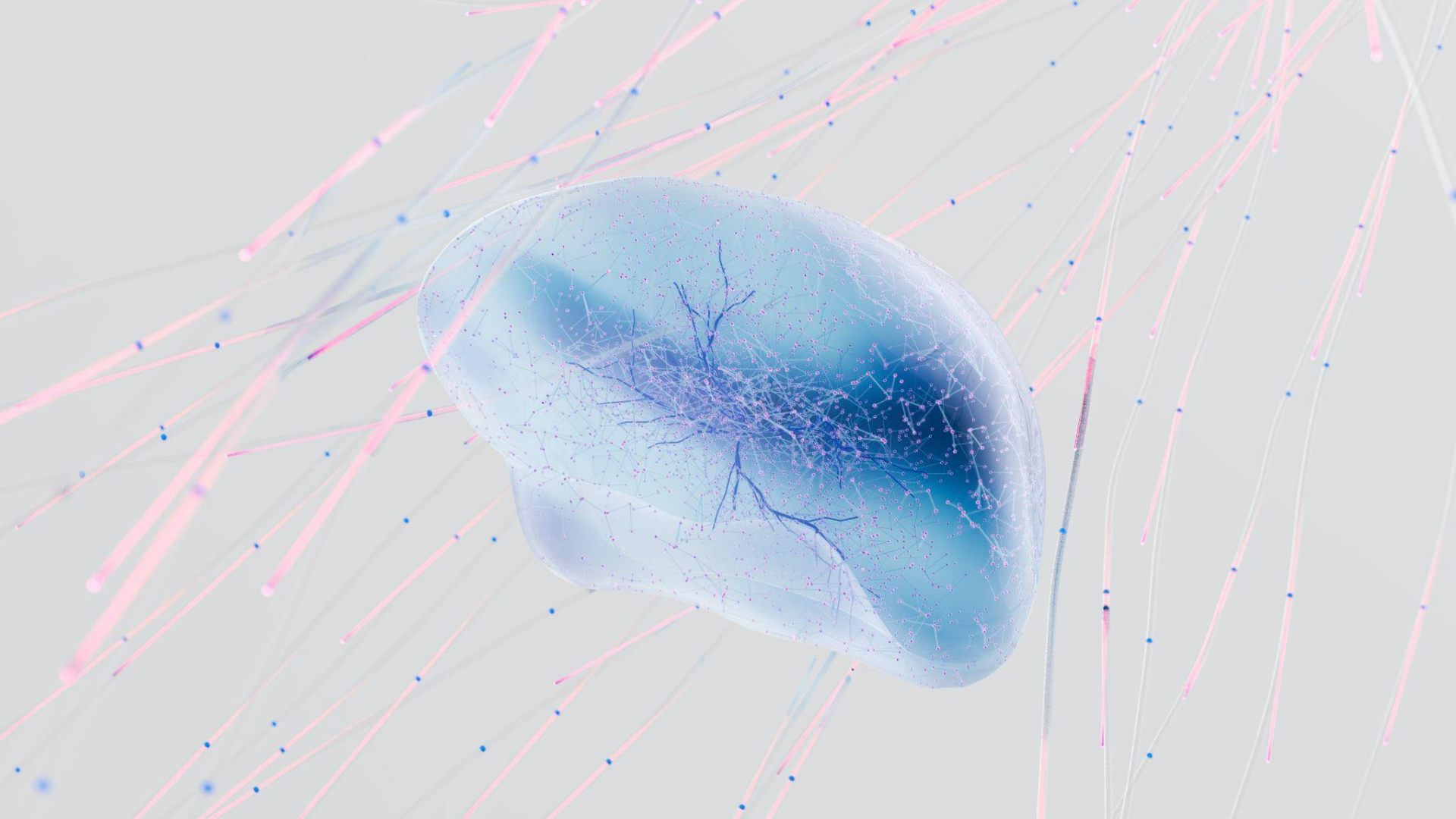
DeepMind

NashLLMs

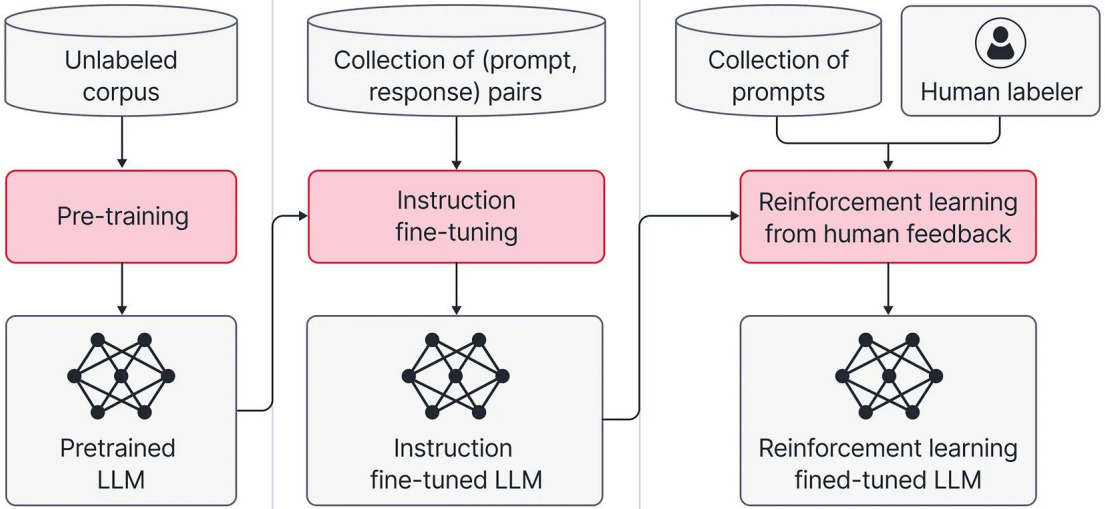
Rémi Munos



Plan for February 67st, 2024

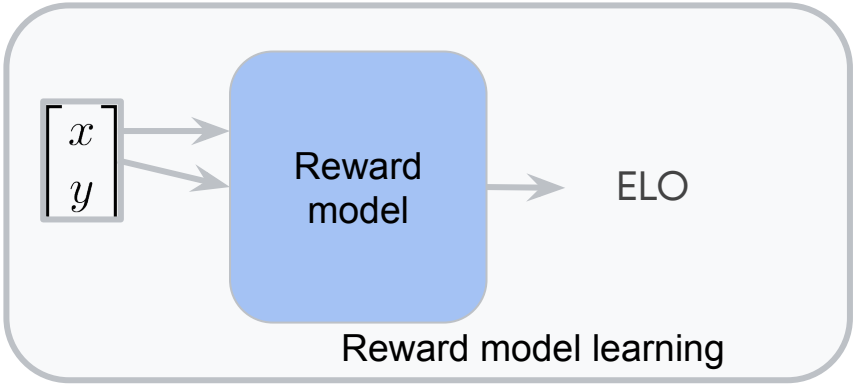
- Algorithmic alignment
- Pairwise preference over ELO scores
- Better than best response
- NashLLMs
- Discussion, Qs, What's next?

Traditional three phases recipe

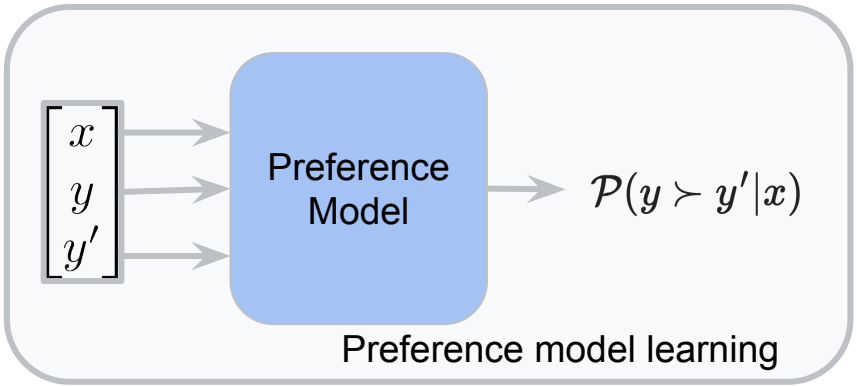


Dataset size	x00 billions to 1.x trillion tokens	~xk to x0k (prompt, response)	~x0k prompt
Example of models	GPT-3, LLaMA, Falcon, BLOOM	Dolly-v2, Falcon-instruct	Claude, GPT-4, ChatGPT
	(a)Pre-training	(b) Instruction fine-tuning	(c) Reinforcement learning from human feedback

Pairwise preference over ELO scores



$$\mathbb{E}_{(y_w,y_l)\sim\mu}\left[f\left(r_\phi(y_w)-r_\phi(y_l)\right)\right]$$



Learn a preference model $\mathcal{P}(y \succ y'|x)$

- Initialise it with a LLM prompted:
  "Given this prompt 'x' and two responses 'y1' and 'y2', which one do you prefer?"
- Trained by SL with preference human data

Best response vs. probability of winning

$$\text{antisymmetric: } \mathcal{P}(y \succ y'|x) = 1 - \mathcal{P}(y' \succ y|x)$$

f is a (deterministic) absolute scoring function

$$\mathcal{P}(y \succ y'|x) = \mathbb{E}_{Z\sim\nu}\left[\mathbb{I}\{f(x, y, Z) \succ f(x, y', Z)\}\right]$$

Probability of winning:

$$\mathcal{P}(\pi \succ \pi'|x) \stackrel{\text{def}}{=} \mathbb{E}_{y\sim\pi(\cdot|x), y'\sim\pi'(\cdot|x)}\left[\mathcal{P}(y \succ y'|x)\right]$$

Best response vs. probability of winning

Probability of winning

$$\mathcal{P}(\pi \succ \pi' | x) \overset{\text{def}}{=} \mathbb{E}_{y \sim \pi(\cdot|x), y' \sim \pi'(\cdot|x)} \left[\mathcal{P}(y \succ y' | x) \right]$$

Nash Equilibrium

$$\arg \max_{\pi} \min_{\pi'} \mathbb{E}_{x, y \sim \pi, y' \sim \pi'} \left[\mathcal{P}(y \succ y' | x) \right]$$

Best response vs. probability of winning

$\mathcal{P}(y \succ y')$	$y = y_1$	$y = y_2$	$y = y_3$
$y' = y_1$	1/2	9/10	2/3
$y' = y_2$	1/10	1/2	2/11
$y' = y_3$	1/3	9/11	1/2

- Can be captured by **BT:** R(y1) = 0, R(y2) = log 9, and R(y3) = log 2
- Unconstrained optimization for maximum reward: **y2 = (0, 1, 0)**
- Unconstrained optimization for best preference: **y2 = (0, 1, 0)**
- Constrained π(y1) = 2π(y2) for maximum reward: **(2/3, 1/3, 0) = P**
- Constrained π(y1) = 2π(y2) for best preference: **(0, 0, 1) = R**

Best response vs. probability of winning

$\mathcal{P}(y \succ y')$	$y = y_1$	$y = y_2$	$y = y_3$
$y' = y_1$	1/2	9/10	2/3
$y' = y_2$	1/10	1/2	2/11
$y' = y_3$	1/3	9/11	1/2

$$\mathbb{E}_{y \sim \pi_R^*}[R(y)] = 0 \times 2/3 + \log(9) \times 1/3 > \log(2) = \mathbb{E}_{y \sim \pi_{\mathcal{P}}^*}[R(y)]$$

$$\mathcal{P}(\pi_{\mathcal{P}}^* \succ \pi_R^*) = \mathcal{P}(y_3 \succ y_1) \times 2/3 + \mathcal{P}(y_3 \succ y_2) \times 1/3 = 50/99 > 1/2$$

Even for BT: “best response” and “probability of winning” differ!

Why stray away from Bradley Terry

1. Diverse human preferences

Example:

- 3 types of humans with respective preferences P1 , P2, P3
- Each type as has a different preference between action y1, y2, y3
- **BT** will select one action y1 deterministically
- **Nash** will selected a mixture policy proportionally

BT is also unstable: One datapoint can radically change the policy

Why stray away from Bradley Terry

2. Limited expressivity

Non transitivity

Example: **Non-transitive dice** (Gardner, 1970)

- We construct: P(π1 > π2) > ½, P(π2 > π3) > ½ , P(π3 > π1) > ½
- π1 = U({2, 4, 9}), π2 = U({1, 6, 8}), and π3 = U({3, 5, 7})

$$\mathcal{P}(\pi_1 \succ \pi_2) = \mathcal{P}(\pi_2 \succ \pi_3) = \mathcal{P}(\pi_3 \succ \pi_1) = 5/9$$

BT is also nonaditive: Bertrand et al. (2023)

Why stray away from Bradley Terry

3. Sensitivity to the sampling distribution

A reward model depends on the data distribution:

$$r^{\pi} \overset{\text{def}}{=} \arg\max_{r(\cdot,\cdot)} \mathbb{E}_{\substack{x \sim \rho \\ y,\, y' \sim \pi(\cdot|x) \\ Z \sim \nu}} \left[\log\left(\sigma(r(x, y_w^Z) - r(x, y_l^Z))\right)\right]$$

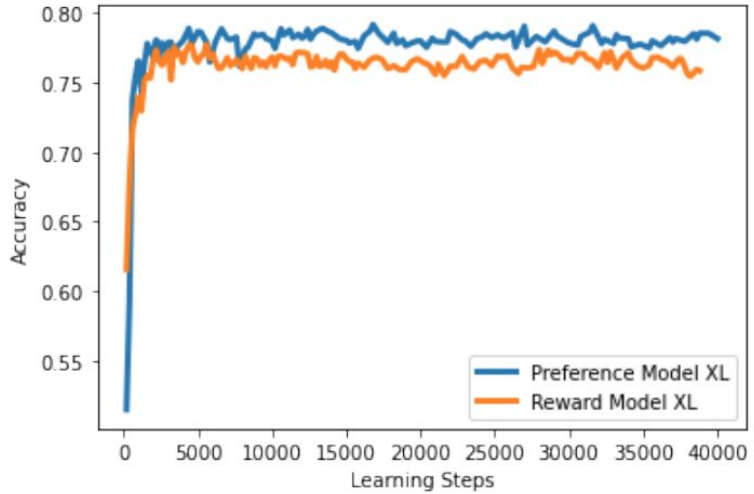
Whereas a preference model essentially* does not:

$$\mathcal{P}^* \overset{\text{def}}{=} \arg\max_{\mathcal{P}(\cdot \succ \cdot|\cdot)} \mathbb{E}_{\substack{x \sim \rho \\ y \sim \pi(\cdot|x) \\ y' \sim \pi'(\cdot|x) \\ Z \sim \nu}} \left[\log \mathcal{P}(y_w^Z \succ y_l^Z|x)\right]$$

essentially* = infinite amount of data, no approximation

Why stray away from Bradley Terry

4. Data comes from human pairwise preferences



Empirical argument: **fits better**

DeepMind

NashLLMs

NashLLM: Preference-based policy gradient for RLHF

NashLLM

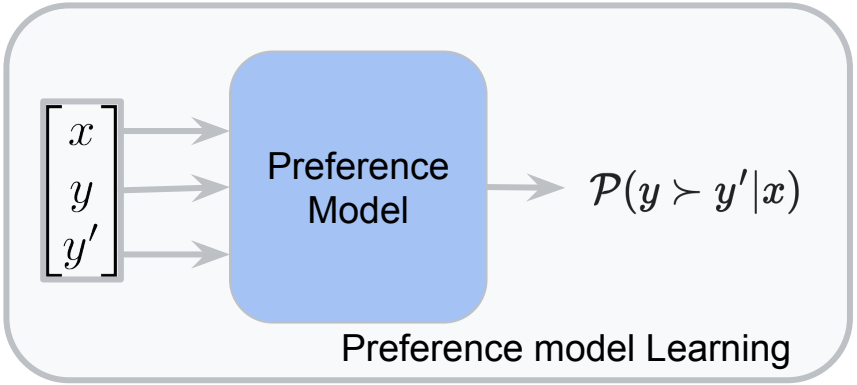
Google DeepMind

Nash Learning from Human Feedback

Rémi Munos[*,1], Michal Valko[*,1], Daniele Calandriello[*,1], Mohammad Gheshlaghi Azar[*,1], Mark Rowland[*,1], Daniel Guo[*,1], Yunhao Tang[*,1], Matthieu Geist[*,1], Thomas Mesnard[1], Andrea Michi[1], Marco Selvi[1], Sertan Girgin[1], Nikola Momchev[1], Olivier Bachem[1], Daniel J. Mankowitz[1], Doina Precup[1] and Bilal Piot[*,1]

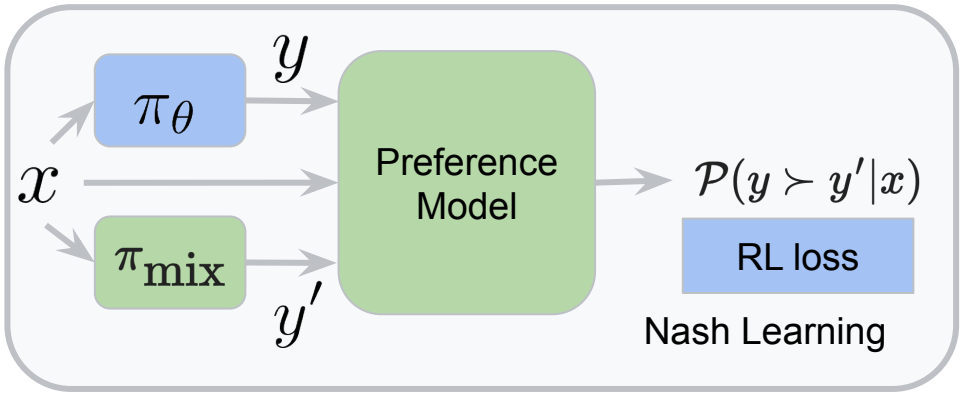
[*]Equal contributions, [1]Google DeepMind

NashLLM: Nash Learning from Human Feedback



Learn a preference model $\mathcal{P}(y \succ y'|x)$

- Initialise it with a LLM prompted: "Given this prompt 'x' and two responses 'y1' and 'y2', which one do you prefer?"
- Trained by SL with preference human data



Compute the Nash equilibrium

$$\arg\max_{\pi}\min_{\pi'} \mathbb{E}_{x,y\sim\pi,y'\sim\pi'}\left[\mathcal{P}(y \succ y'|x)\right]$$

- Find policy that generates responses preferred over alternative policies
- **Nash-MD algorithm:** improve by playing against a mixture between current and past policies

Back to NashLLM: RLHF vs NLHF algorithmically

NashLLM

RLHF with PPO / DPO / ... work with ELO score
↝ maximises the *expected reward*

$$\nabla \log \pi(a|x)\Big(R - V(x)\Big)$$

We are after: Policy preferred by humans

💡 **New criterion:** Maximise the **probability of producing a preferred answer**

$$\arg\max_{\pi} \min_{\pi'} \mathbb{E}_{x,y\sim\pi,y'\sim\pi'}\left[\mathcal{P}(y \succ y'|x)\right]$$

🌳 **Unexpected benefit:** Variance reduction for free!

NashLLM: Addressing reward hacking

The regularized preference between actions $y \sim \pi(\cdot|x), y' \sim \pi'(\cdot|x)$ is defined as

$$\mathcal{P}_\tau^{\pi,\pi'}(y \succ y'|x) \stackrel{\text{def}}{=} \mathcal{P}(y \succ y'|x) - \tau \log \frac{\pi(y|x)}{\mu(y|x)} + \tau \log \frac{\pi'(y'|x)}{\mu(y'|x)},$$

and we define accordingly the KL-regularized preference between policies:

$$\begin{aligned} \mathcal{P}_\tau(\pi \succ \pi') &\stackrel{\text{def}}{=} \mathbb{E}_{x\sim\rho, y\sim\pi(\cdot|x), y'\sim\pi'(\cdot|x)}\left[\mathcal{P}_\tau^{\pi,\pi'}(y \succ y'|x)\right] \\ &= \mathcal{P}(\pi \succ \pi') - \tau \mathrm{KL}_\rho(\pi, \mu) + \tau \mathrm{KL}_\rho(\pi', \mu), \end{aligned}$$

🌳 **Unexpected benefits:**

- Regularized NE is unique!
- Can get fast convergence in distribution!
- Get last iterate convergence!

NashLLM: Self-improvement

🌳 Construct the preference model giving pairwise reward

$$R(x, a, a') = \mathbb{P}_{h\sim\mathcal{H}}(\text{human } h \text{ prefers } a \text{ over } a'|x)$$

🌳 Compute the Nash equilibrium

$$\arg\max_{\pi}\min_{\pi'} \mathcal{P}(\pi > \pi'), \quad \text{where:} \quad \mathcal{P}(\pi > \pi') = \mathbb{E}_{x,a\sim\pi,a'\sim\pi'}[R(x, a, a')]$$

Step 1: Given the base policy π_0 find a preferred policy π_1

Step 2: Given policies π_0 and π_1 find a policy π_2 preferred over π_0 and π_1

Step 3: Given π_0 and π_1 and π_2 find a policy π_3 preferred over π_0 and π_1 and π_2

...

End: Finds a policy π_{NASH} preferred over all

NashMD in LLMs

🌳 Full NashMD asks for **best-response** (BR) in every step

$$\pi_{t+1} = \arg\max_{\pi} \left[\eta \mathcal{P}(\pi \succ \pi_t) - \mathrm{KL}(\pi, \pi_t^{\mu})\right]$$

🌳 **NashMD**-PG: follow the gradient - note the difference in the KL!

$$\nabla_\theta \log \pi_\theta(y|x) \left[\mathcal{P}(y \succ y'|x) - \frac{1}{2}\right] - \tau \nabla_\theta \mathrm{KL}(\pi_\theta(\cdot|x), \pi_{ref}(\cdot|x))$$

🌳 **y** is generated from the current policy
🌳 **y'** is generated from a (geometric) mixture between the current policy and a past checkpoint (such as the initial SFT policy):

$$y' \sim \pi_\theta^\beta(\cdot|x) \propto (\pi_\theta(\cdot|x))^{1-\beta} (\pi_{ref}(\cdot|x))^\beta$$

Experiment on a text summarizing task

Train preference model (T5X-L models) on TL;DR database, then compute the Nash using several methods: Self-Play, Nash-MD, Nash-EMA, Best-Response.

Table 1. PaLM 2 preference $\mathcal{P}^*(\pi_c \succ \pi_r)$ model between column policy π_c against row policy π_r.

$\mathcal{P}^*$	SFT	RLHF	SP	**MD1**	MD2	MD3	MD4	MD5	MD6	BR	EMA1	EMA2	EMA1*	EMA2*
SFT	0.500	0.990	0.983	**0.982**	0.989	0.987	0.985	0.982	0.965	0.943	0.970	0.961	0.977	0.980
RLHF	0.010	0.500	**0.489**	**0.598**	**0.519**	**0.561**	**0.501**	**0.436**	**0.284**	**0.148**	0.468	0.320	0.477	0.510
SP	0.017	0.511	0.500	**0.592**	0.504	0.545	0.499	0.451	0.310	0.211	0.445	0.362	0.464	0.488
MD1	0.018	0.402	0.408	**0.500**	0.425	0.470	0.369	0.362	0.238	0.163	0.391	0.270	0.400	0.447
MD2	0.011	0.481	0.496	**0.575**	0.500	0.513	0.491	0.434	0.298	0.196	0.460	0.351	0.430	0.496
MD3	0.013	0.439	0.455	**0.530**	0.487	0.500	0.484	0.408	0.273	0.187	0.429	0.323	0.413	0.472
MD4	0.015	0.499	0.501	**0.631**	0.509	0.516	0.500	0.428	0.265	0.161	0.468	0.358	0.437	0.503
MD5	0.018	0.564	0.549	**0.638**	0.566	0.592	0.572	0.500	0.329	0.210	0.532	0.389	0.518	0.539
MD6	0.035	0.716	0.690	**0.762**	0.702	0.727	0.735	0.671	0.500	0.342	0.652	0.548	0.651	0.691
BR	0.057	0.852	0.789	**0.837**	0.804	0.813	0.839	0.790	0.658	0.500	0.743	0.640	0.752	0.774
EMA1	0.030	0.532	0.555	**0.609**	0.540	0.571	0.532	0.468	0.348	0.257	0.500	0.381	0.480	0.556
EMA2	0.039	0.680	0.638	**0.730**	0.649	0.677	0.642	0.611	0.452	0.360	0.619	0.500	0.585	0.659
EMA1*	0.023	0.523	0.536	**0.600**	0.570	0.587	0.563	0.482	0.349	0.248	0.520	0.415	0.500	0.555
EMA2*	0.020	0.490	0.512	**0.553**	0.504	0.528	0.497	0.461	0.309	0.226	0.444	0.341	0.445	0.500

https://arxiv.org/abs/2312.00886

Many more open questions!

Michal Valko
https://misovalko.github.io/

- Offline/IPO-ish NashMD
- Online IPO / dependent data distribution
- Join SFT + RL fine-tuning
- Alignment in pretraining already
- IPO Robustification (adversarial alignment)
- Adapting fast fine-tuning and retuning
- Non-linear trajectory reward for fine-grained HF
- General non-pairwise, conversational