Sequential Decision Making: Theory and Practice

Lecture 9: RLHF

Yuejie Chi

Yale Statistics & Data Science



Fall 2026

Outline

Introduction to LLM posttraining

RLHF

DPO

NLHF

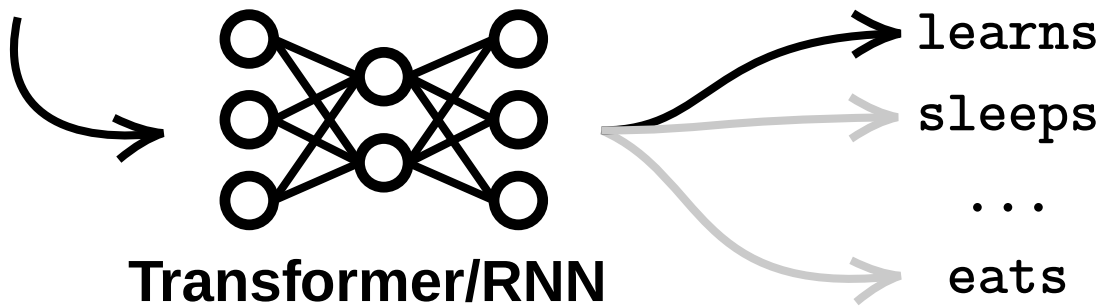
Best-of-N alignment

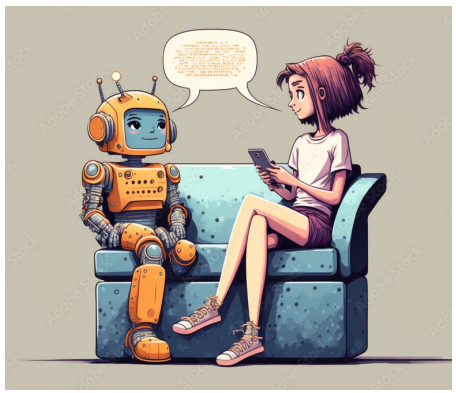
Large language models

The language models predict the next word based on previous words.

Prompt: Explain reinforcement learning (RL).

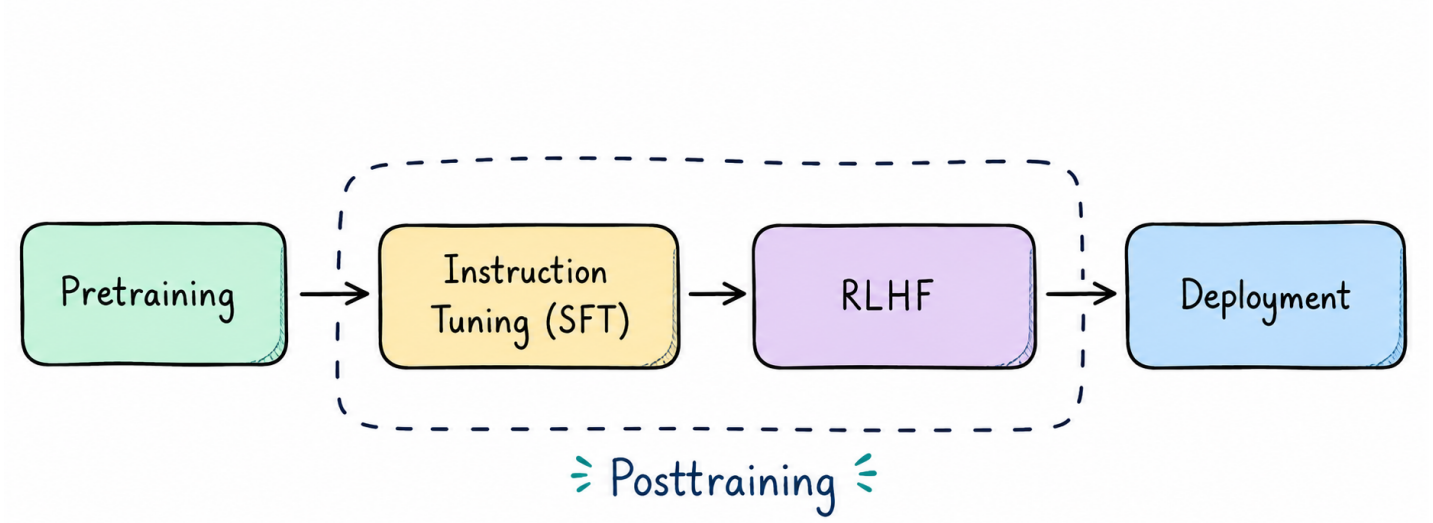
Answer: Reinforcement learning (RL) is a type of machine learning where an agent





LLMs can do amazing things beyond their original conception.

From pretraining to posttraining



- Pretraining: train on large corpus of text data using next-token prediction.
- Instruction tuning: finetuning on a collection of tasks formatted with instruction
- RLHF: finetuning to align with human preferences

Why posttraining? smaller models with targeted posttraining often outperform larger generalist models in their trained domains of expertise.

Instruction finetuning

Instruction tuning aligns a pretrained LLM to follow human instructions by learning from prompt-response examples.

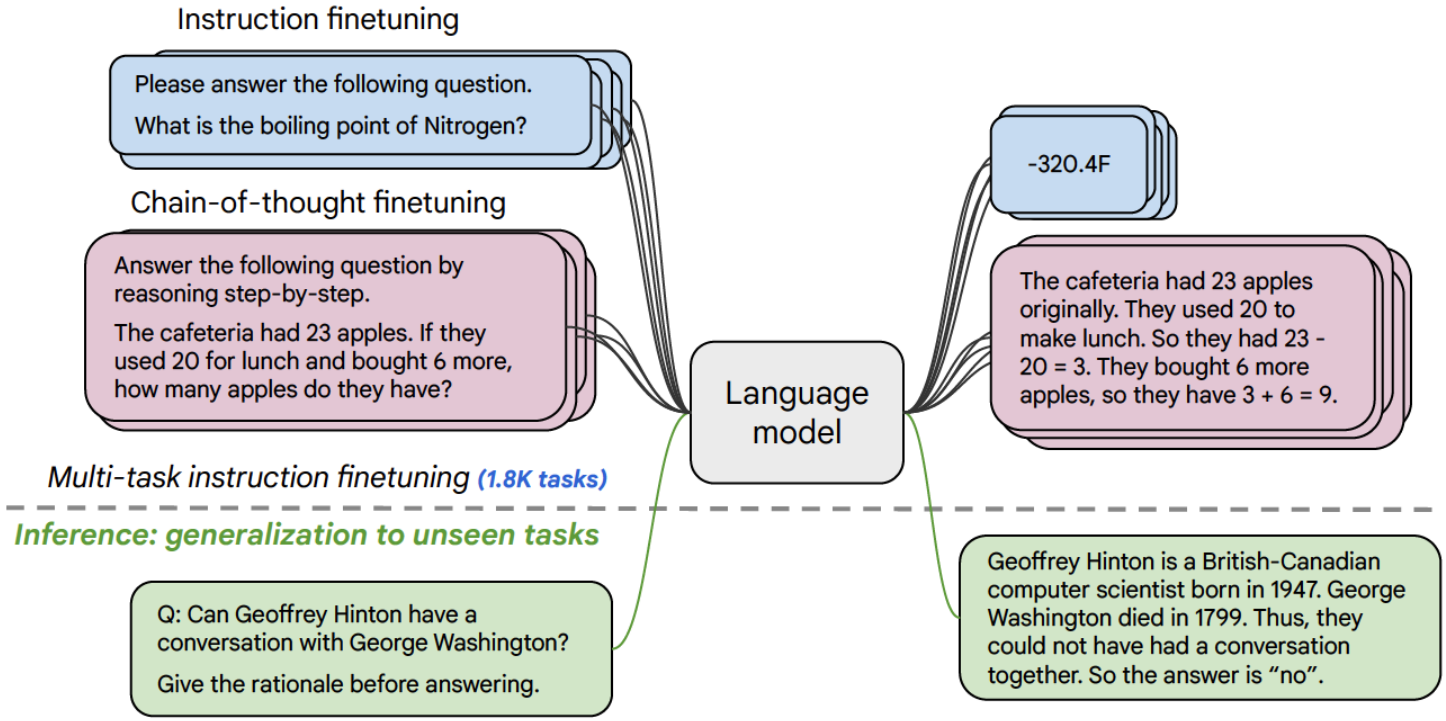


Figure credit: [Chung et al., 2024]

Before and after instruction finetuning

Model learns to follow instructions and adapts to new tasks.

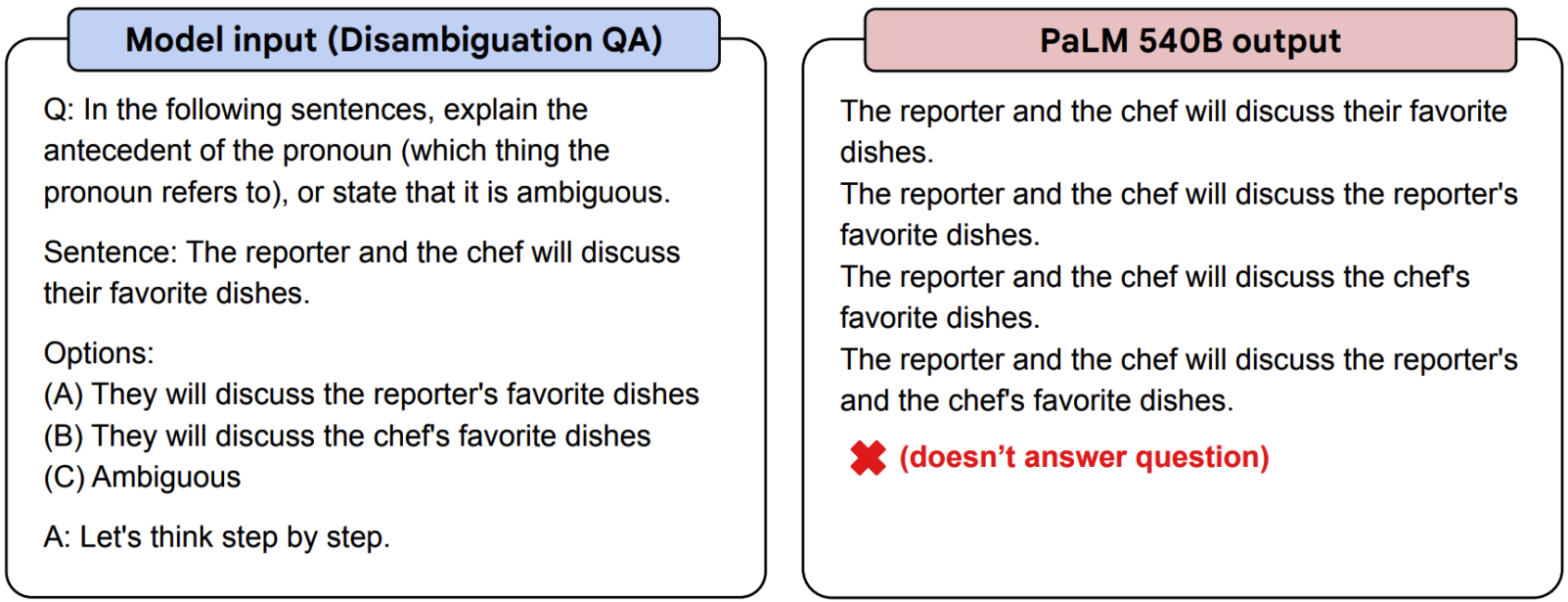


Figure credit: [Chung et al., 2024]

Before and after instruction finetuning

Model learns to follow instructions and adapts to new tasks.

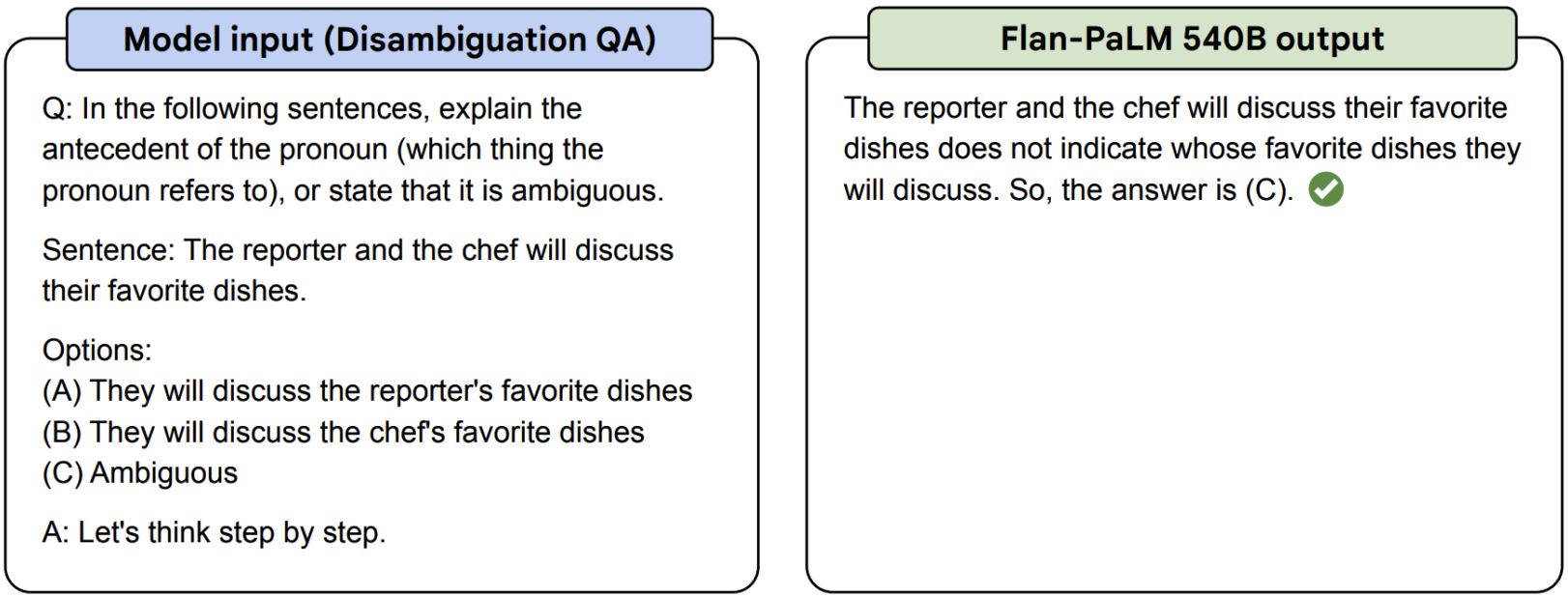


Figure credit: [Chung et al., 2024]

Limitations of instruction finetuning

Can you think of any?

- It's expensive to curate the ground-truth data for tasks: human versus LLM-generated data.
- Open-ended creative generation have no right answer.
- Degradation of base model performance and hallucination.
- Not consistent with human preference: helpful, truthful, and safety.

Next: finetune the LLM to align with human preference

RLHF

Language models as policies

Sentence-level: Given prompt $x \in \mathcal{X}$, a language model generates an answer:

$$y \sim \underbrace{\pi(\cdot|x)}_{\text{parameterized by LLM}}$$

- **Contextual bandit**: the prompt x is the context.

Token-level: given all the tokens up to time t, an *auto-regressive* language model generates the next token:

$$x_t \sim \pi(\cdot|x_{<t})$$

- **Deterministic MDP**: $x_{<t}$ is the state, and the next state is given by concatenating state $x_{<t}$ and action x_t.

We focus on the sentence-level model in the lecture.

The reward model

A **reward model** measures how well an answer y is to prompt x according to human preference:

$$r(x, y)$$

We would like to have

$$y_1 \succ y_2 \mid x \quad \iff \quad r(x, y_1) > r(x, y_2).$$

The RLHF pipeline

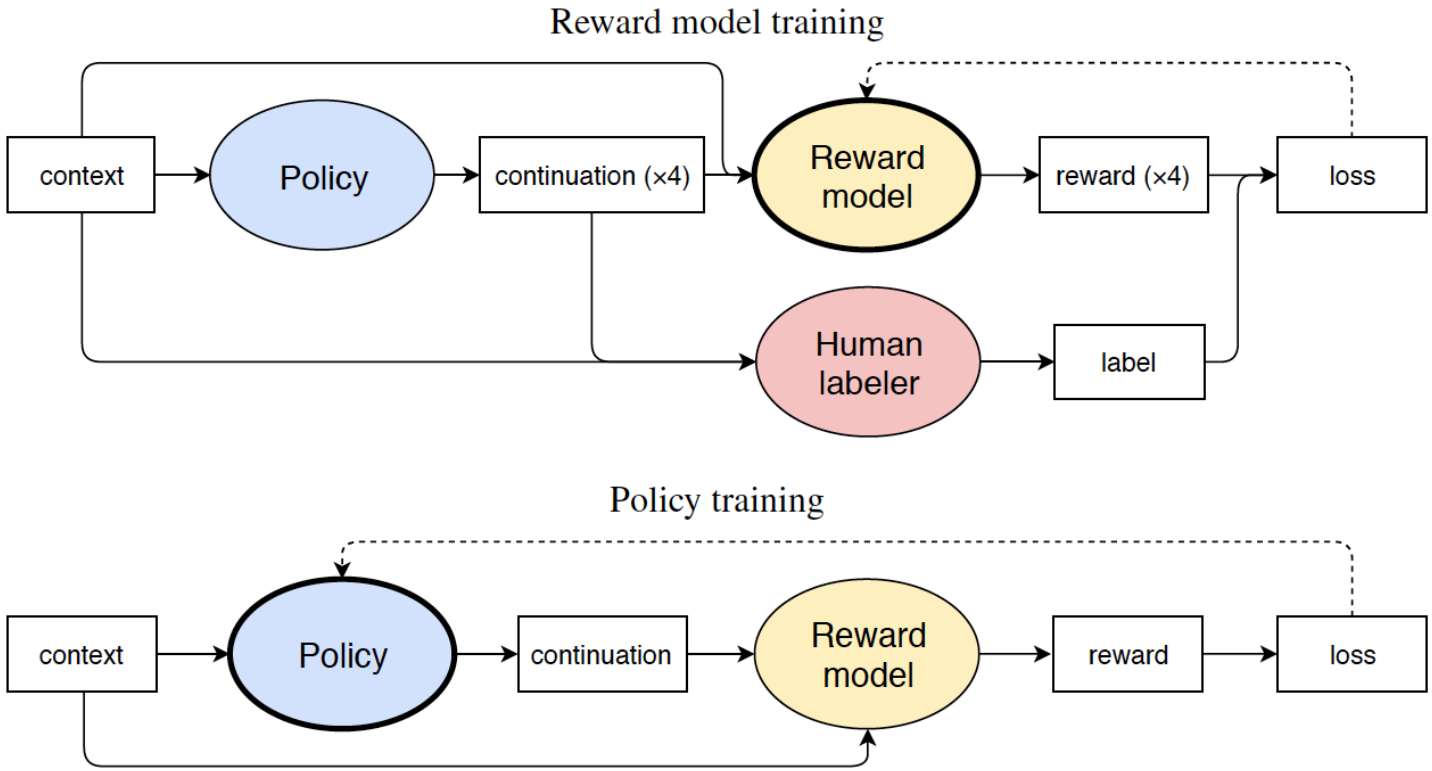
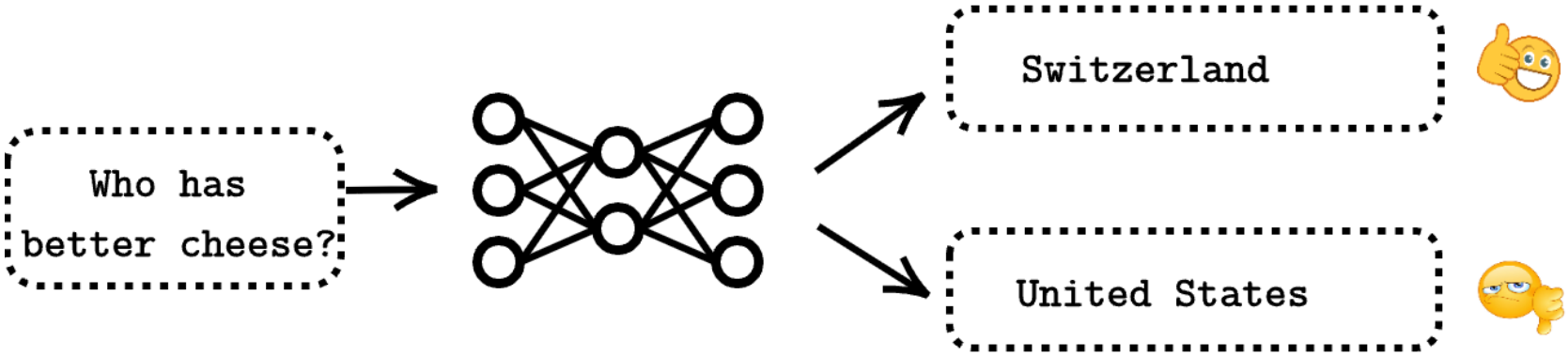


Figure credit: [Ziegler et al., 2019]

- *Reward learning*: learn a reward model from preference data;
- *Policy optimization*: optimize the LLM to maximize the reward.

Preference data from pairwise comparisons



- Why pairwise comparison data?
- What about other types of data?
  - thumb up/down, absolute scores, categorical scores, K-way comparison...

Ranking model

Bradley-Terry model

The probability of pairwise comparison $i \succ j$ is modeled by

$$\mathbb{P}(i \succ j) = \frac{\exp(r_i^\star)}{\exp(r_i^\star) + \exp(r_j^\star)} = \sigma(r_i^\star - r_j^\star),$$

where $r_i^\star \in \mathbb{R}$ is the score associated with item i.

- **Reward model:** $r^\star : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, evaluating the quality of a prompt-answer pair (x, y) that aligns with human preference;
- BT model cannot distinguish between r and $r + c \cdot \mathbf{1}$.
- More general ranking model can be used, such as K-way comparisons via softmax.

RLHF: Reward learning

Comparison data: i.i.d. data tuples

$$\mathcal{D} = \{(x^i, y_+^i, y_-^i)\}_{i=1}^N$$

Maximum-likelihood estimation: given comparison data $\mathcal{D}$, the MLE of the reward function is given by

$$r_{\mathsf{MLE}} = \operatorname*{argmin}_r \ell(r, \mathcal{D}),$$

where

$$\ell(r, \mathcal{D}) = -\sum_{i=1}^N \log \sigma(r(x^i, y_+^i) - r(x^i, y_-^i)).$$

The binary cross-entropy loss encourages larger rewards to preferred answers.

RLHF: policy optimization

Policy optimization via reward maximization

Find π that (approximately) maximizes the objective w.r.t. r

$$J(r,\pi) = \underbrace{\mathop{\mathbb{E}}_{x\sim\rho, y\sim\pi(\cdot|x)}[r(x,y)]}_{\textit{reward}} - \beta \underbrace{\mathop{\mathbb{E}}_{x\sim\rho}[\mathsf{KL}(\pi(\cdot|x)\,\|\,\pi_{\mathrm{ref}}(\cdot|x))]}_{\textit{diversity/pretraining}}$$

$$= \mathop{\mathbb{E}}_{x\sim\rho, y\sim\pi(\cdot|x)}\left[r(x,y) - \beta \log \frac{\pi(\cdot|x)}{\pi_{\mathrm{ref}}(\cdot|x)}\right]$$

- $\beta > 0$: KL regularization parameter;
- π_{ref}: a reference policy, typically the model after SFT;
- $\rho \in \Delta(\mathcal{X})$: prompt distribution.

Often optimized via PPO [Schulman et al., 2017] in practice (we'll discuss PPO next lecture).

Reward hacking

KL regularization is important to prevent reward hacking.

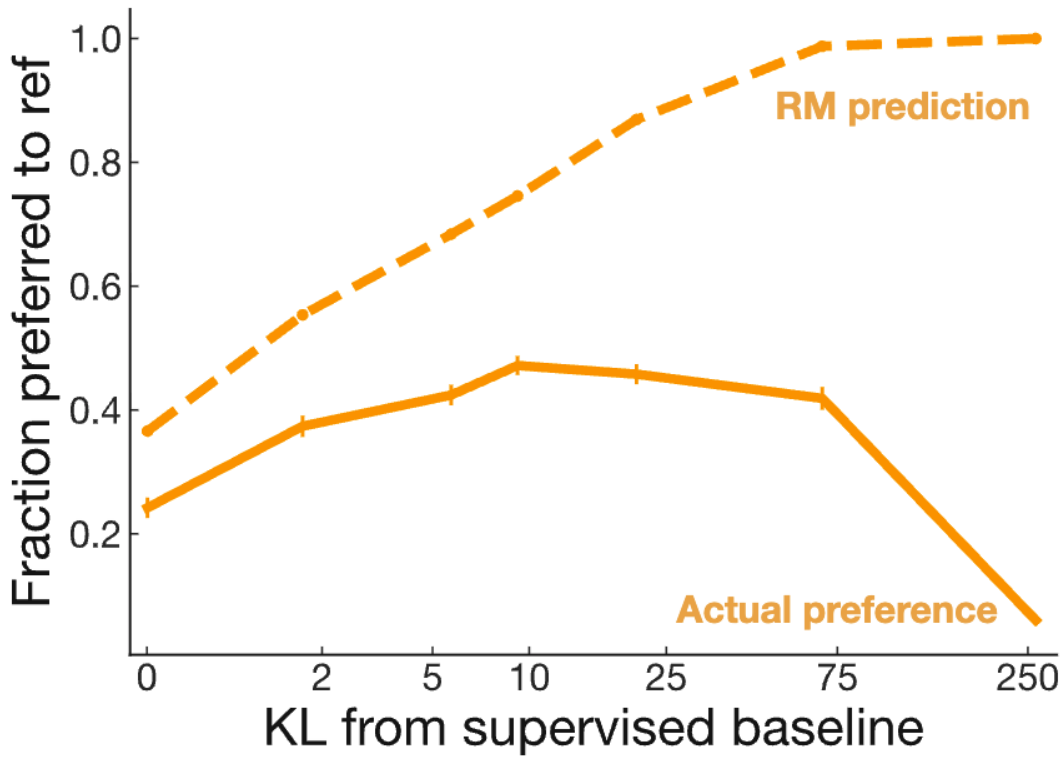


Figure credit: [Stiennon et al., 2020]

Pareto frontier

$$r_{\text{reg}}(x, y) = r(x, y) - \beta \log \frac{\pi(\cdot|x)}{\pi_{\text{ref}}(\cdot|x)}$$

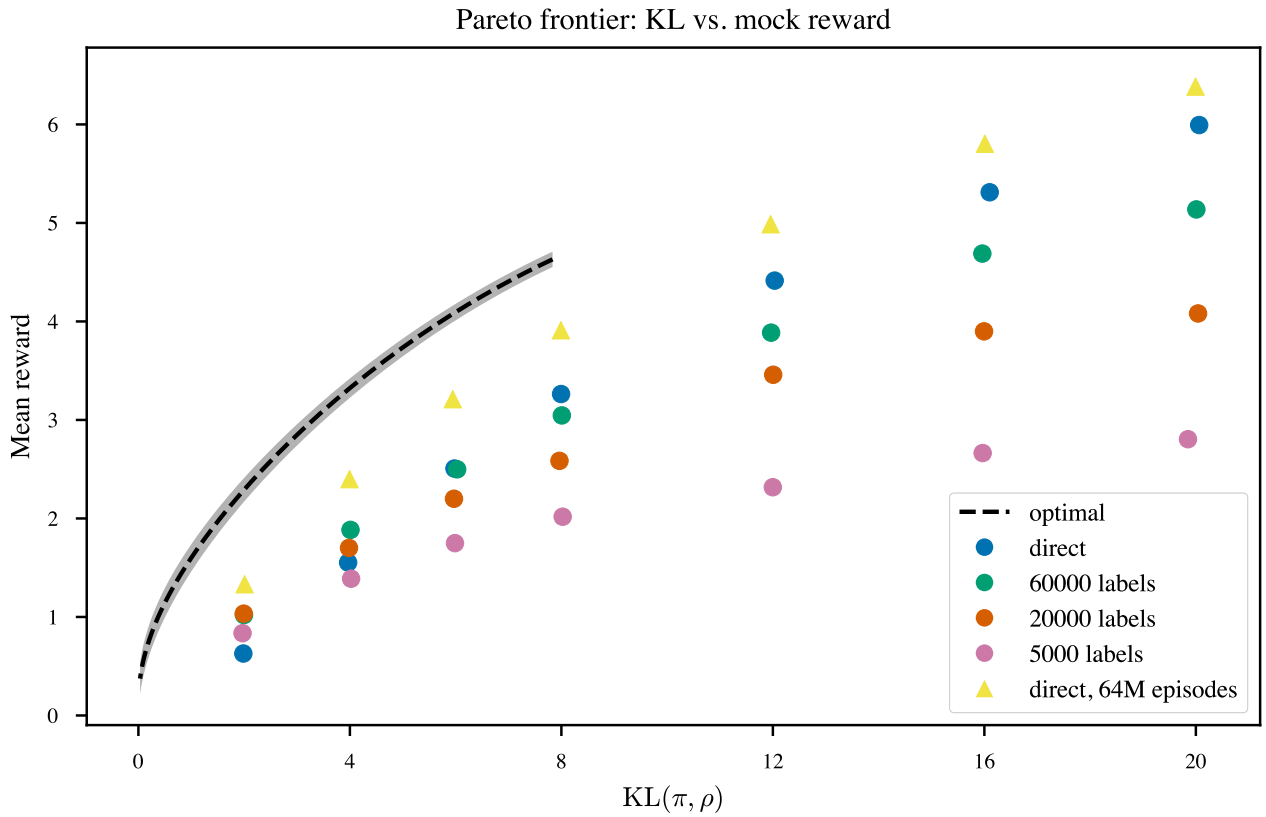


Figure credit: [Ziegler et al., 2019]

InstructGPT

An early milestone which applies SFT and RLHF to GPT-3.

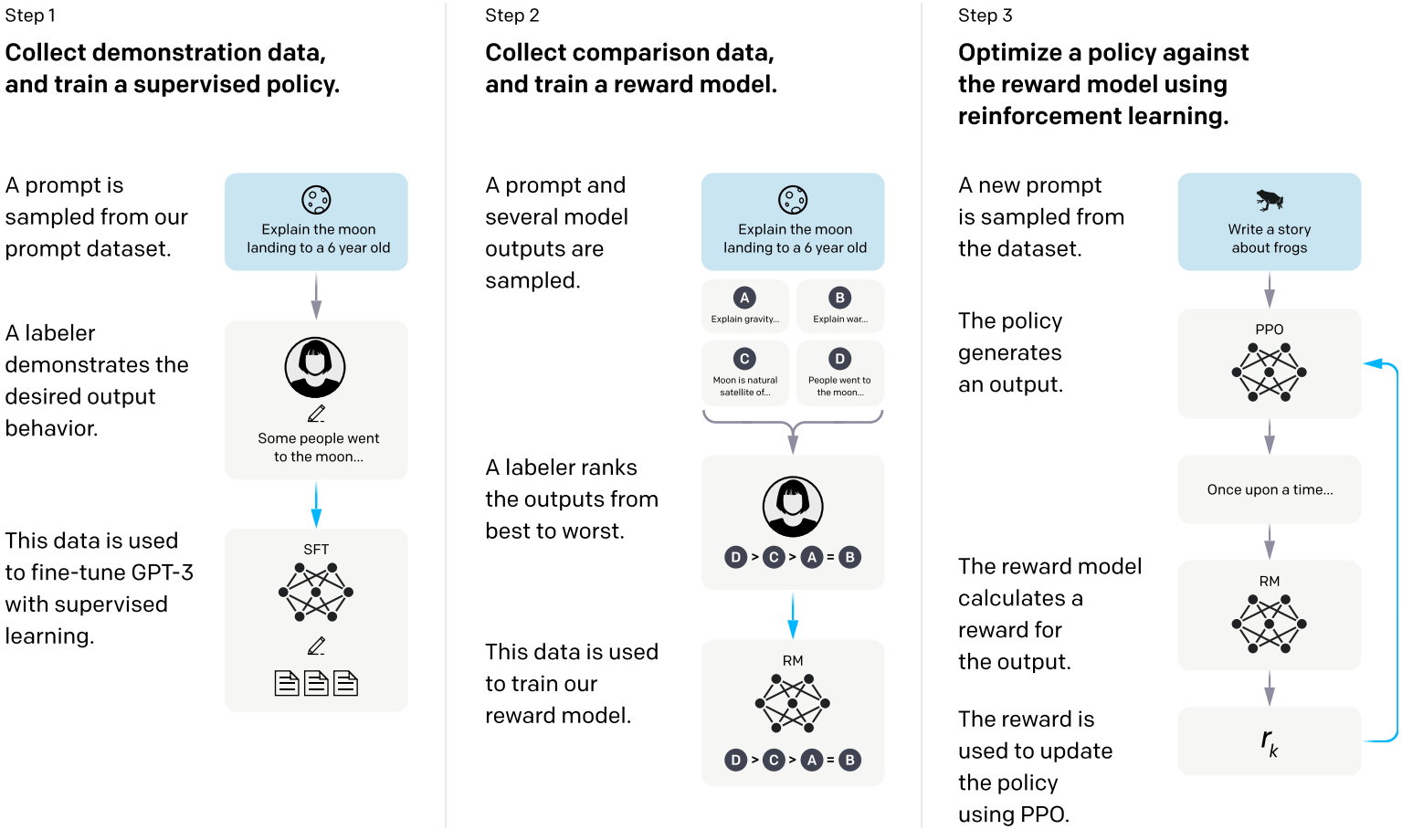


Figure credit: [Ouyang et al., 2022]

InstructGPT: performance gains

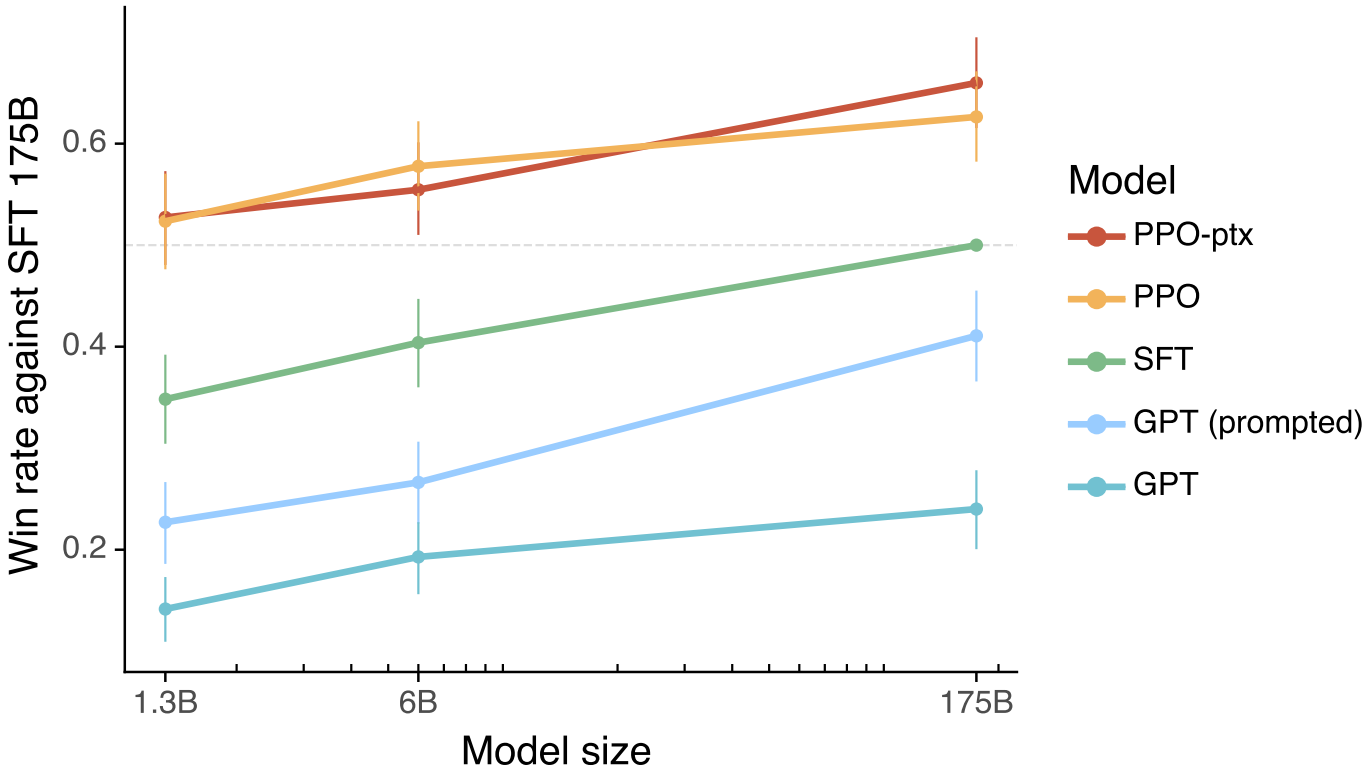


Figure credit: [Ouyang et al., 2022]

Take-away: the smaller model with RLHF (at 1.3B) can outperform bigger models (at 175B) without RLHF.

Direct Preference Optimization

Direct Preference Optimization (DPO)

— [Rafailov et al., 2023]

1. Reward learning: $\widehat{r} \leftarrow \operatorname{argmin}_r \ell(r, \mathcal{D})$
2. Policy learning: $\widehat{\pi} \leftarrow \operatorname{argmax}_\pi J(\widehat{r}, \pi)$

Key observation: for step 2, the optimal π w.r.t. r admits a closed-form solution

$$\pi_r = \operatorname*{argmax}_\pi J(r, \pi) \iff \pi_r(y|x) = \frac{\pi_{\text{ref}}(y|x) \exp(r(x,y)/\beta)}{Z(r,x)}.$$

Proof of the closed-form optimal policy

For any policy π,

$$
\begin{aligned}
J(r,\pi) &= \mathop{\mathbb{E}}_{x\sim\rho,\, y\sim\pi(\cdot|x)}\left[r(x,y) - \beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)}\right] \\
&= -\beta \mathop{\mathbb{E}}_{x\sim\rho,\, y\sim\pi(\cdot|x)}\left[\log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)\exp\big(r(x,y)/\beta\big)}\right] \\
&= -\beta \mathop{\mathbb{E}}_{x\sim\rho,\, y\sim\pi(\cdot|x)}\left[\log \frac{\pi(y|x)}{\pi_r(y|x)Z(r,x)}\right] \\
&= \underbrace{\beta \mathop{\mathbb{E}}_{x\sim\rho}\left[\log Z(r,x)\right]}_{\text{independent of } \pi} - \beta \mathop{\mathbb{E}}_{x\sim\rho}\left[\mathsf{KL}\big(\pi(\cdot|x)\,\|\,\pi_r(\cdot|x)\big)\right].
\end{aligned}
$$

Therefore,

$$
J(r,\pi_r) - J(r,\pi) = \beta \mathop{\mathbb{E}}_{x\sim\rho}\left[\mathsf{KL}\big(\pi(\cdot|x)\,\|\,\pi_r(\cdot|x)\big)\right] \geq 0.
$$

Direct Preference Optimization (DPO)

— [Rafailov et al., 2023]

- The reward function r in terms of its optimal π_r is

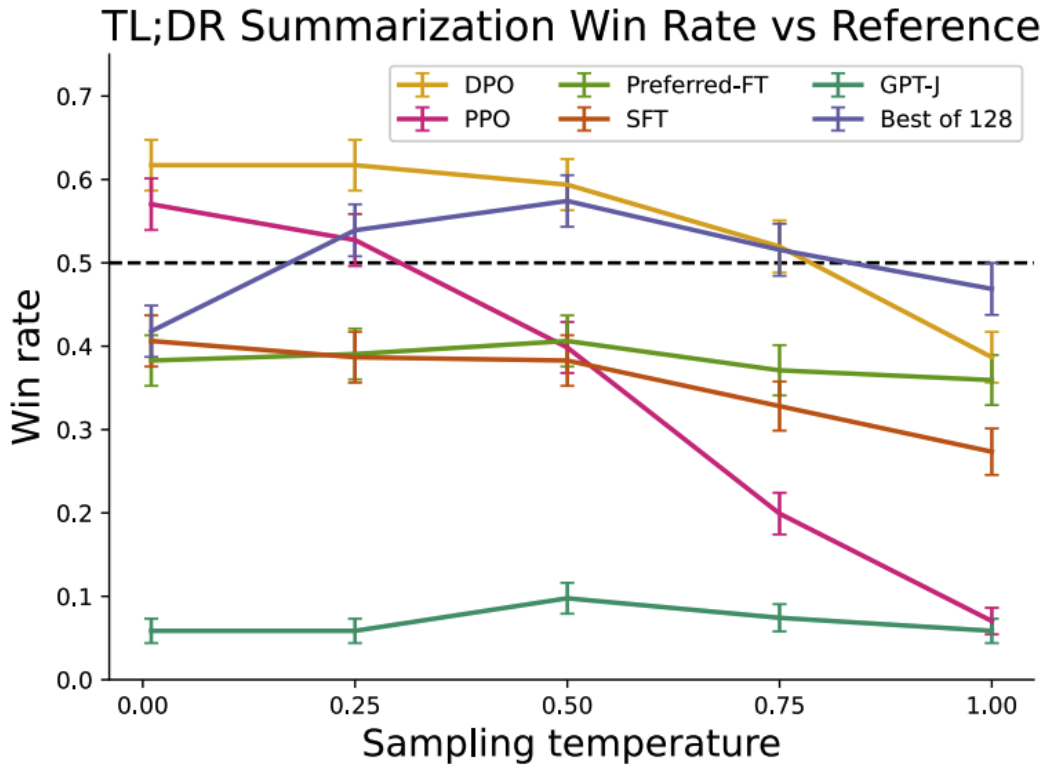
$$r(x, y) = \underbrace{\beta\big(\log \pi_r(y|x) - \log \pi_{\text{ref}}(y|x) + \log Z(r, x)\big)}_{=:r(\pi)}.$$

- The two-step procedure is equivalent to

$$\begin{aligned}\widehat{\pi} &\leftarrow \operatorname*{argmin}_{\pi} \ell(r(\pi), \mathcal{D}) \\ &= \operatorname*{argmin}_{\pi} -\sum_{i=1}^{N} \log \sigma \left(\beta \log \frac{\pi(y_+^i|x^i)}{\pi_{\text{ref}}(y_+^i|x^i)} - \beta \log \frac{\pi(y_-^i|x^i)}{\pi_{\text{ref}}(y_-^i|x^i)} \right).\end{aligned}$$

Single-step and **policy-only**! Very popular in practice.

Performance of DPO

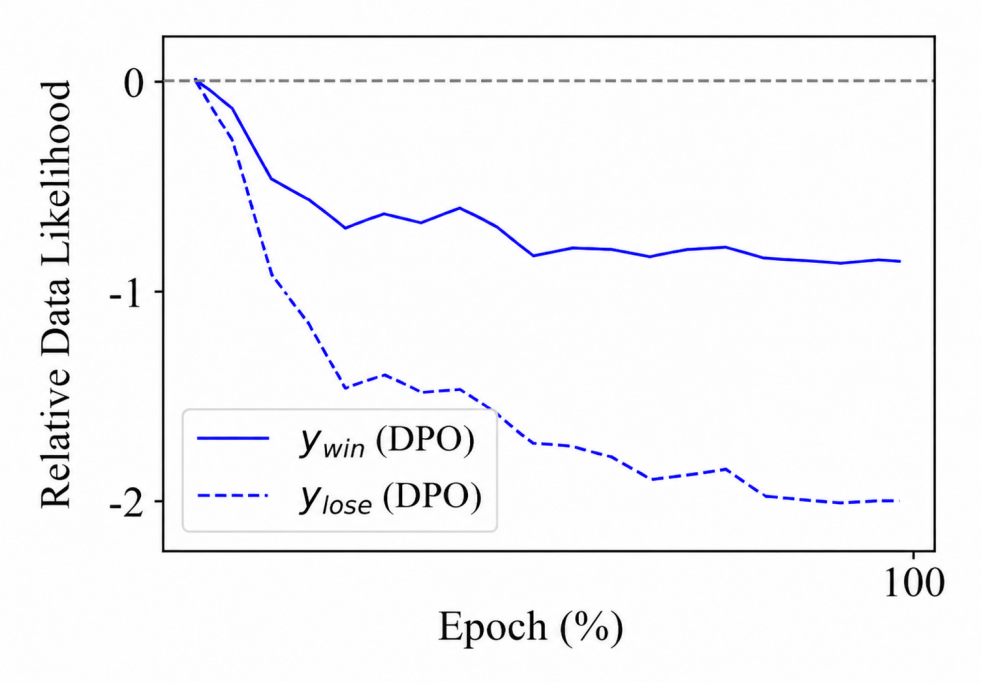


TL;DR GPT-4 evaluated summarization win rates versus human-written summaries: DPO performs well without explicit reward modeling.

Failure mode of DPO

DPO tends to decrease both $\pi(y_+|x)$ and $\pi(y_-|x)$ in training, while having their ratio increase:

$$\widehat{\pi} = \underset{\pi}{\operatorname{argmin}} - \sum_{i=1}^{N} \log \sigma \left(\beta \log \frac{\pi(y_+^i|x^i)}{\pi(y_-^i|x^i)} - \beta \log \frac{\pi_{\text{ref}}(y_+^i|x^i)}{\pi_{\text{ref}}(y_-^i|x^i)} \right).$$



A toy illustration of the failure mode

Consider a toy setting [Pal et al., 2024, Razin et al., 2025]:

- 3 possible answers: "`Yes`", "`No`", "`Never`"
- We have a single pair of data: $(x, y_+, y_-) = (x, \texttt{No}, \texttt{Never})$.
- The unobserved incorrect answer $y_0 = \texttt{Yes}$.

Response	Initial policy	New policy
$y_+ = \texttt{No}$	0.4	0.3
$y_- = \texttt{Never}$	0.4	0.1
$y_0 = \texttt{Yes}$	0.2	0.6
$\frac{\pi(y_+\|x)}{\pi(y_-\|x)}$	1	3

Preference ratio improves, but probability of a correct answer decreases.

- **Shared representation couple the responses:** suppressing y_- may suppress y_+.

Mitigation: DPO with an NLL term

Possible fix: for each preference pair (x, y_+, y_-), augment the DPO loss [Pang et al., 2024]:

$$\ell_{\text{DPO+NLL}}(\theta) = \ell_{\text{DPO}}(\theta) + \lambda \underbrace{\left[-\frac{\log \pi_\theta(y_+|x)}{|y_+|} \right]}_{\text{negative log-likelihood (NLL)}} .$$

- $|y_+|$: response length;
- $\lambda > 0$: regularization parameter.
- The NLL term pushes up the preferred answer's log-likelihood.

Many more variants of DPO in the literature!

Nash Learning from Human Feedback

Drawbacks of scalar rewards

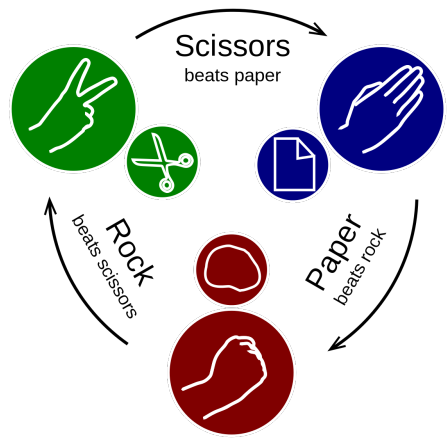
The scalar reward model assigns one score per response, and imposes a transitive ordering:

$$y_1 \succ y_2 \mid x \quad \iff \quad r(x, y_1) > r(x, y_2).$$

Lack of transitivity: population preferences can be cyclic:

$$y_1 \succ y_2, \qquad y_2 \succ y_3, \qquad y_3 \succ y_1.$$

No scalar score can represent all three comparisons simultaneously.



A scalar reward cannot fully represent a general population preference.

Preference model

A preference model predicts comparisons directly rather than passing through a scalar score.

Game-theoretic formulation: the preference/payoff model captures the *win probability* of response y over y' for prompt x:

$$P_x(y, y') := \mathbb{P}(y \succ y' \mid x).$$

The matrix P_x can retain cycles and heterogeneity in human preferences.

From reward to win rates: a reward model induces a payoff matrix:

$$P_x^r(y, y') = \mathbf{1}\{r(x, y) > r(x, y')\} + \underbrace{\tfrac{1}{2}\mathbf{1}\{r(x, y) = r(x, y')\}}_{\text{random tie-breaking}}.$$

Therefore, the payoff-based model is a richer model.

Nash learning from human feedback (NLHF)

Win rate: for policies π and π', define their expected pairwise preference

$$P(\pi \succ \pi') = \mathbb{E}_{x\sim\rho,\, y\sim\pi(\cdot|x),\, y'\sim\pi'(\cdot|x)}[P_x(y, y')]$$
$$= \mathbb{E}_{x\sim\rho}\underbrace{\left[\pi(\cdot|x)^\top P_x \pi'(\cdot|x)\right]}_{\text{matrix game}}.$$

This is called the *win rate* of π over π'.

Win rate dominance via self-play [Munos et al., 2024]

$$\pi^\star \in \operatorname*{argmax}_{\pi} \min_{\pi'} P(\pi \succ \pi').$$

- For fixed π, $\min_{\pi'} P(\pi \succ \pi')$ measures its smallest win rate against any policy.
- No competing policy beats $\pi^\star$ in expectation:

$$P(\pi \succ \pi^\star) \leq \tfrac{1}{2} \quad \text{for every competing policy } \pi.$$

KL-regularized game

Let π_{ref} be a reference policy and $\tau > 0$. NLHF considers the KL-regularized game [Munos et al., 2024]

$$P_\tau(\pi \succ \pi') = P(\pi \succ \pi') - \tau \mathsf{KL}(\pi \| \pi_{\text{ref}}) + \tau \mathsf{KL}(\pi' \| \pi_{\text{ref}}).$$

- Regularization yields a unique Nash equilibrium (the QRE).
- τ controls preference improvement versus distribution shift.

Optimal policy characterization

We have the following fixed point characterization of the optimal policy:

$$\begin{aligned}\pi_\tau^\star &= \operatorname*{argmax}_\pi \min_{\pi'} P_\tau(\pi \succ \pi') \\ &= \operatorname*{argmax}_\pi \{P(\pi \succ \pi_\tau^\star) - \tau \mathsf{KL}(\pi \| \pi_{\text{ref}})\}\end{aligned}$$

Solving the regularized preference game

Optimal policy characterization

We have the following **fixed-point** characterization of the optimal policy:

$$\begin{aligned}\pi_\tau^\star &= \operatorname*{argmax}_\pi \min_{\pi'} P_\tau(\pi \succ \pi') \\ &= \operatorname*{argmax}_\pi \{P(\pi \succ \pi_\tau^\star) - \tau \mathsf{KL}(\pi \| \pi_{\mathrm{ref}})\}\end{aligned}$$

Many algorithms have been developed to solve the KL-regularized matrix game with *last-iterate* convergence, as discussed in Lecture 7:

- Policy mirror descent [Munos et al., 2024];
- optimistic multiplicative weights (OMWU) [Cen et al., 2022].

The last-iterate convergence means the current policy itself approaches equilibrium, which is critical for LLM applications.

Best-of-N alignment

Alignment at inference time: Best-of-N

Can we improve alignment of a frozen model at inference time?

Best-of-N (BoN) sampling: given a prompt x, a reference policy π_{ref}, and a reward model r:

1. Draw N independent candidates:

$$y_1, \ldots, y_N \overset{\text{iid}}{\sim} \pi_{\text{ref}}(\cdot|x).$$

2. Score the candidates and return the highest-scoring response:

$$i^\star \in \operatorname*{argmax}_{1 \le i \le N} r(x, y_i), \qquad y^{\text{BoN}} = y_{i^\star}.$$

   Break ties uniformly among the maximizing candidates.

Benefit: one model can serve different rewards without changing the weights.

BoN policy

We can think of the BoN sampling as a policy:

$$y^{\mathrm{BoN}} \sim \pi_{\mathrm{BoN}}(\cdot|x)$$

Question: what is the BoN policy, and how does it improve over π_{ref}, say, in terms of win rate?

Recall: a reward model induces a payoff matrix:

$$P_x^r(y, y') = \mathbf{1}\{r(x, y) > r(x, y')\} + \underbrace{\tfrac{1}{2}\mathbf{1}\{r(x, y) = r(x, y')\}}_{\text{random tie-breaking}}.$$

Then the win rate of a policy π over the reference policy is:

$$P^r(\pi \succ \pi_{\mathrm{ref}}) = \mathbb{E}_{x\sim\rho,\, y\sim\pi(\cdot|x)} \underbrace{\mathbb{E}_{y'\sim\pi_{\mathrm{ref}}(\cdot|x)} P_x^r(y, y')}_{=:q_{\mathrm{ref}}(x,y)},$$

where $q_{\mathrm{ref}}(x, y)$ is the win probability of answer y for prompt x.

BoN policy

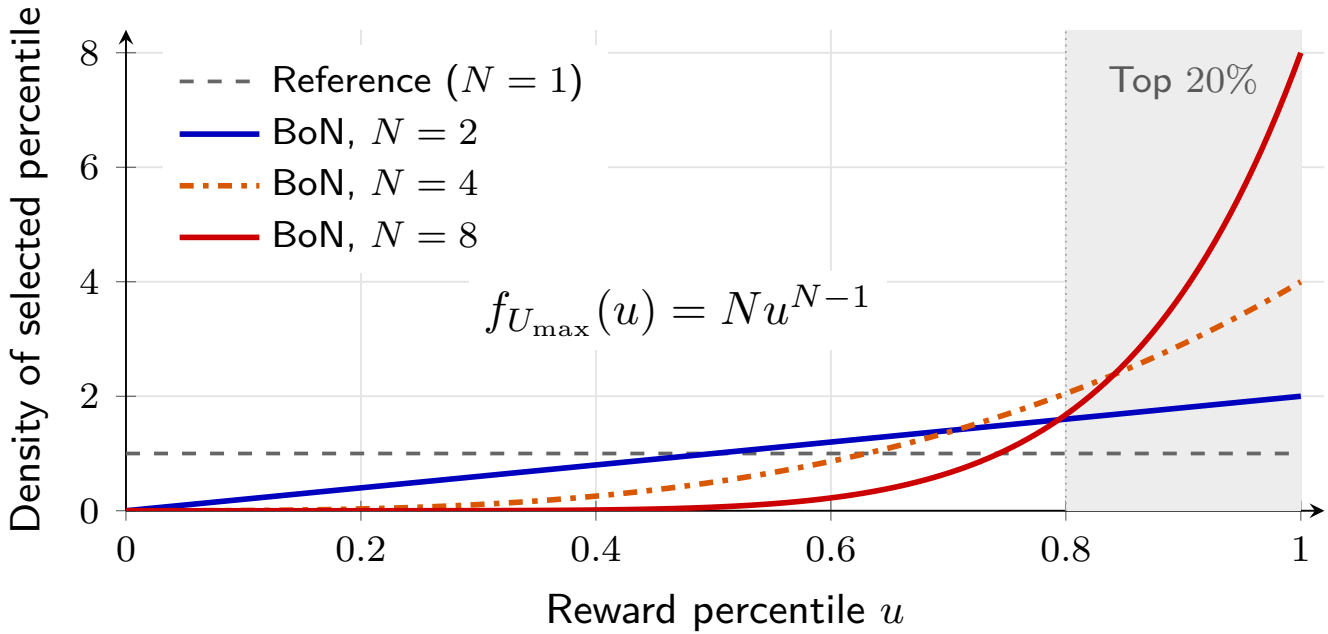
If the reference reward distribution is continuous, the win probability

$$q_{\text{ref}}(x, y) = \mathbb{P}_{Y \sim \pi_{\text{ref}}(\cdot|x)}(r(x, Y) \leq r(x, y))$$

is the response's reward percentile.

The BoN policy [Gui et al., 2024]

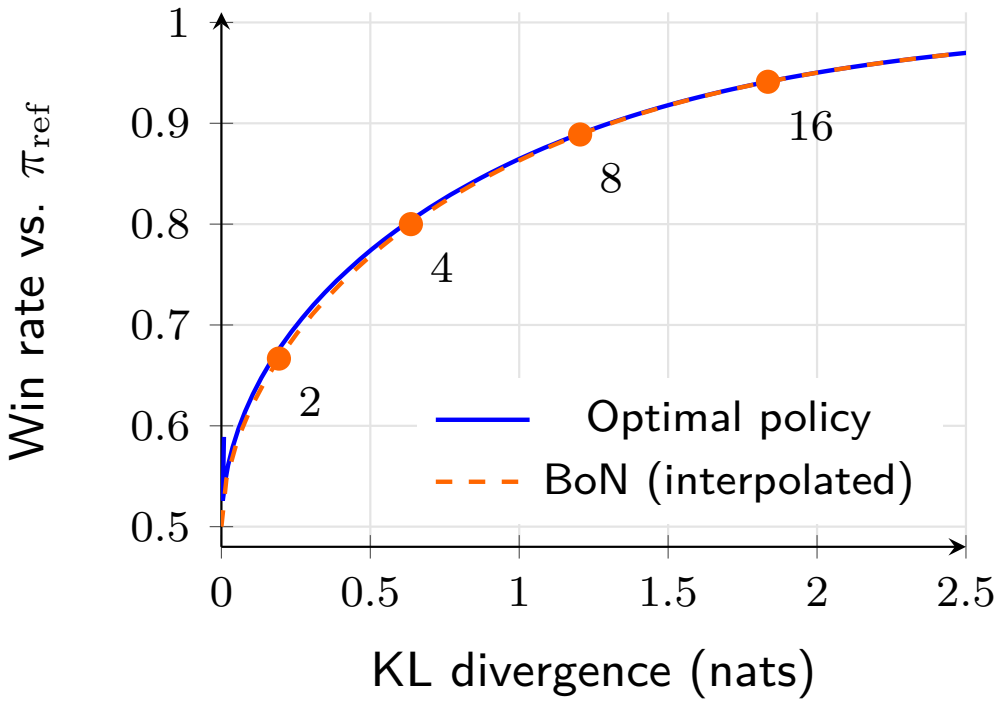
$$\pi_{\text{BoN}}(y|x) := N\, q_{\text{ref}}(x, y)^{N-1} \pi_{\text{ref}}(y|x).$$



The win-rate–KL tradeoff

Under the continuous-reward assumption [Gui et al., 2024]:

$$P^r(\pi_{\text{BoN}} \succ \pi_{\text{ref}}) = \frac{N}{N+1}, \qquad \text{KL}(\pi_{\text{BoN}} \| \pi_{\text{ref}}) = \log N - \frac{N-1}{N}.$$



Optimal policy:

Given KL budget d, maximize the win rate against π_{ref} over all policies. This gives the **best achievable win rate** at each budget.

BoN nearly matches the optimal policy at the same KL budget.

What is the limit of BoN?

Challenge of implementing BoN: the inference cost is high for (relatively) large N

Solution: distillation the BoN policy to serve.

Iterative BoN:

- Get π_{BoN} using π_{ref}
- Replace π_{ref} by π_{BoN}: selecting the best of N independent BoN outputs is equivalent to selecting the best of $N \times N = N^2$ samples from the original policy.
- Repeating this process gives

$$\text{Best-of-}N \longrightarrow \text{Best-of-}N^2 \longrightarrow \text{Best-of-}N^3 \longrightarrow \cdots .$$

The limit achieves win-rate dominance [Yang et al., 2024].

References I



Cen, S., Cheng, C., Chen, Y., Wei, Y., and Chi, Y. (2022).

Fast global convergence of natural policy gradient methods with entropy regularization.

Operations Research, 70(4):2563–2578.



Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. (2024).

Scaling instruction-finetuned language models.

Journal of Machine Learning Research, 25(70):1–53.



Gui, L., Gârbacea, C., and Veitch, V. (2024).

BoNBoN alignment for large language models and the sweetness of best-of-n sampling.

In *Advances in Neural Information Processing Systems*, volume 37.



Munos, R., Valko, M., Calandriello, D., Azar, M. G., Rowland, M., Guo, Z. D., Tang, Y., Geist, M., Mesnard, T., et al. (2024).

Nash learning from human feedback.

In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*.

References II



Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022).

Training language models to follow instructions with human feedback.

arXiv preprint arXiv:2203.02155.



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.



Pang, R. Y., Yuan, W., He, H., Cho, K., Sukhbaatar, S., and Weston, J. E. (2024).

Iterative reasoning preference optimization.

In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.



Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. (2023).

Direct preference optimization: your language model is secretly a reward model.

In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 53728–53741.

References III



Razin, N., Malladi, S., Bhaskar, A., Chen, D., Arora, S., and Hanin, B. (2025).
Unintentional unalignment: Likelihood displacement in direct preference optimization.
In *The Thirteenth International Conference on Learning Representations*.



Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017).
Proximal policy optimization algorithms.
arXiv preprint arXiv:1707.06347.



Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. (2020).
Learning to summarize with human feedback.
Advances in neural information processing systems, 33:3008–3021.



Yang, T., Mei, J., Dai, H., Wen, Z., Cen, S., Schuurmans, D., Chi, Y., and Dai, B. (2024).
Faster WIND: Accelerating iterative best-of-N distillation for LLM alignment.
arXiv preprint arXiv:2410.20727.

References IV



Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019).

Fine-tuning language models from human preferences.

arXiv preprint arXiv:1909.08593.