Lecture 14: RL from Human Feedback
CS885 Reinforcement Learning

2025-02-27

Complementary readings:
Stiennon, Ouyang, Wu, Ziegler, Lowe Voss, Radford, Amodei, Christiano (2020) Learning to summarize from human feedback, NeurIPS.
Ouyang, Wu, Jiang, Wainwright, et al. (2022) Training language models to follow instructions with human feedback, NeurIPS.
Holtzman, Buys, Du, Forbes, Choi (2019). The Curious Case of Neural Text Degeneration, arxiv.
Rafailov, Sharma, Mitchell, Ermon, Manning, Finn (2023) Direct Preference Optimization: Your Language Model is Secretly a Reward Model, NeurIPS.
Rashid, Wu, Fan, Li, Kristiadi, Poupart (2025) Towards Cost-Effective Reward Guided Text Generation, arxiv.

Pascal Poupart
David R. Cheriton School of Computer Science

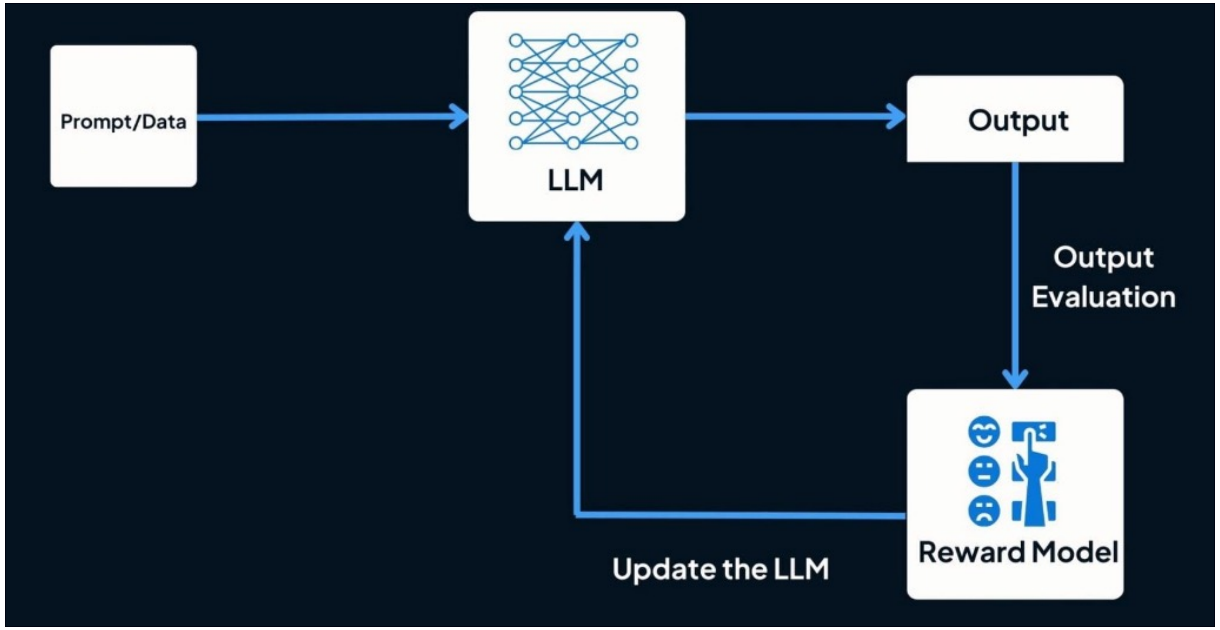
UNIVERSITY OF WATERLOO

Outline

- Reinforcement Learning from Human Feedback
- Direct Preference Optimization
- Reward Guided Text Generation

Large Language Models

- **Agent:** system
- **Environment:** user
- **State:** history of past utterances
- **Action:** system utterance
- **Reward:** task completion, human feedback

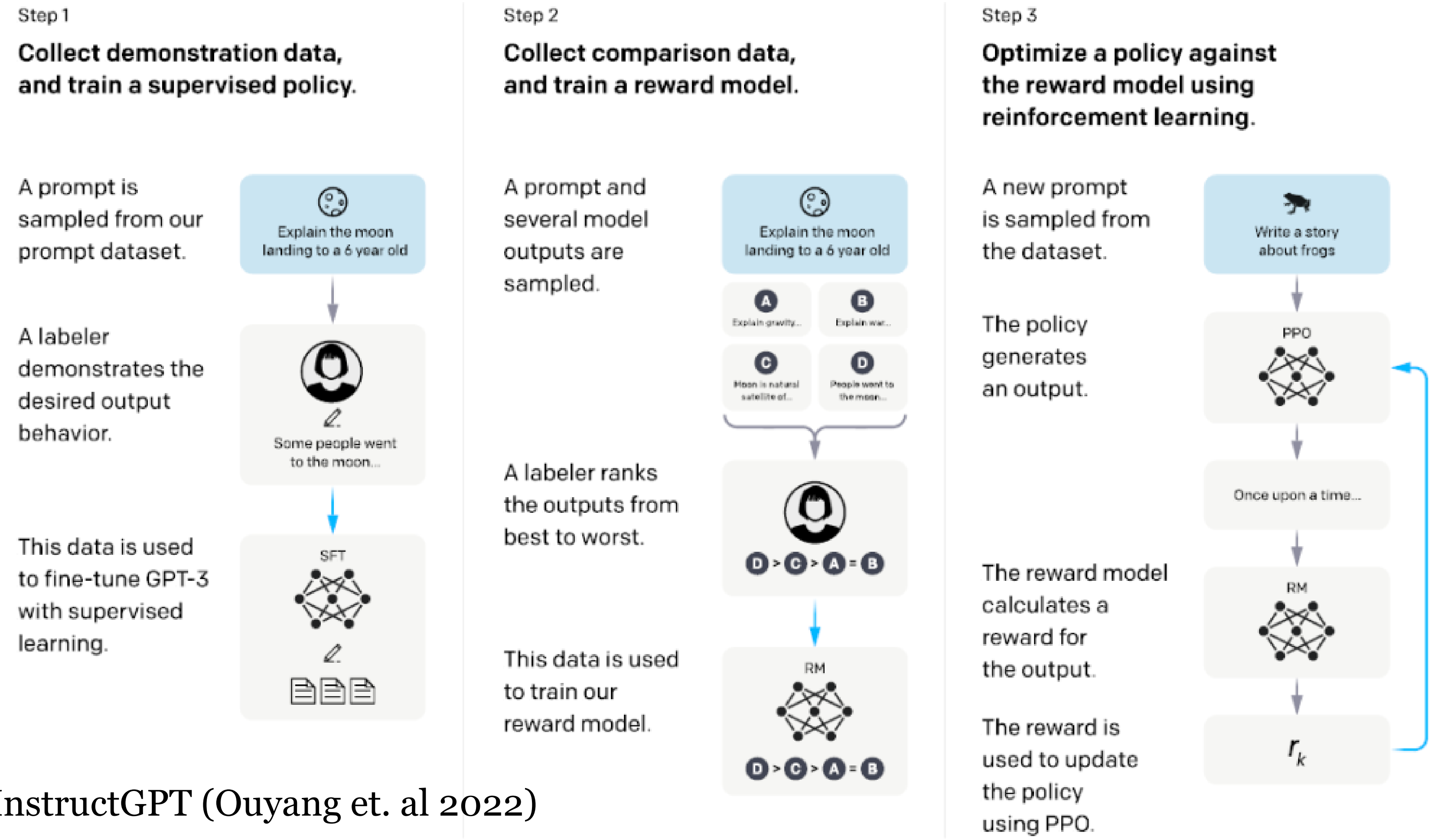


Credit: https://www.twine.net/blog/what-is-reinforcement-learning-from-human-feedback-rlhf-and-how-does-it-work/

"*We posit that the superior writing abilities of LLMs, as manifested in surpassing human annotators in certain tasks, are fundamentally driven by RLHF, as documented in Gilardi et al. (2023) and Huang et al. (2023).*"
Llama 2 Technical Report (Touvron et. al 2023)

"*This behavior (re-evaluation) is not only a testament to the model's growing reasoning abilities but also a captivating example of how reinforcement learning can lead to unexpected and sophisticated outcomes*"
Deepseek-R1 Technical Report (Guo et. al 2025)

Recipe For Large Language Model Training



InstructGPT (Ouyang et. al 2022)

Response Ranking

PP Explain reinforcement learning in three sentences.

‹ 1 / 2 › Reinforcement learning is a type of machine learning where an agent learns to make decisions by receiving feedback in the form of rewards or punishments. The agent's goal is to maximize its cumulative reward over time by exploring different actions and learning which actions lead to the highest rewards. Reinforcement learning has been successfully applied to a variety of tasks, including game playing, robotics, and recommendation systems.

‹ 2 / 2 › Reinforcement learning is a type of machine learning that focuses on training an agent to make decisions based on trial and error. The agent receives feedback in the form of rewards or penalties for each decision it makes. By learning from this feedback, the agent can improve its decision-making abilities over time.

Was this response better or worse? Better Worse Same

RL from Human Feedback (RLHF)

- Collect a preference data set:

$$D = \{(\boldsymbol{s}, \boldsymbol{a}_+, \boldsymbol{a}_-)_k\}_{k=1}^{K} \text{ where } \boldsymbol{a}_+ \succ \boldsymbol{a}_-$$

- Train a reward model according to the Bradley Terry Model:

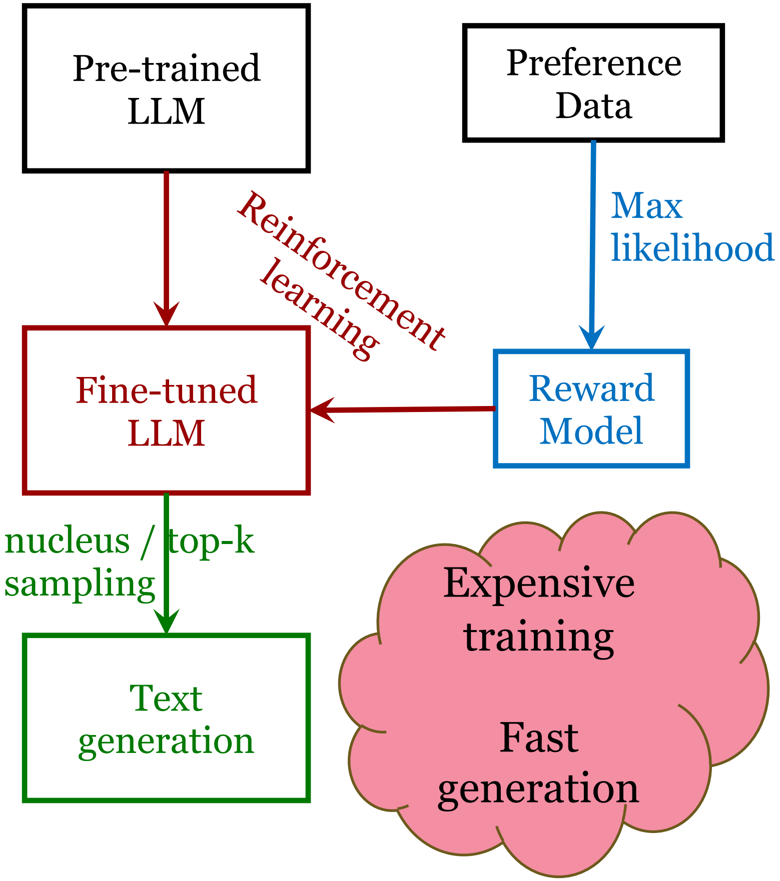
$$\max_{\theta} E_D[\log \sigma(r_\theta(\boldsymbol{s}, \boldsymbol{a}_+) - r_\theta(\boldsymbol{s}, \boldsymbol{a}_-))]$$

- Make a copy of the LLM and finetune it to maximize:

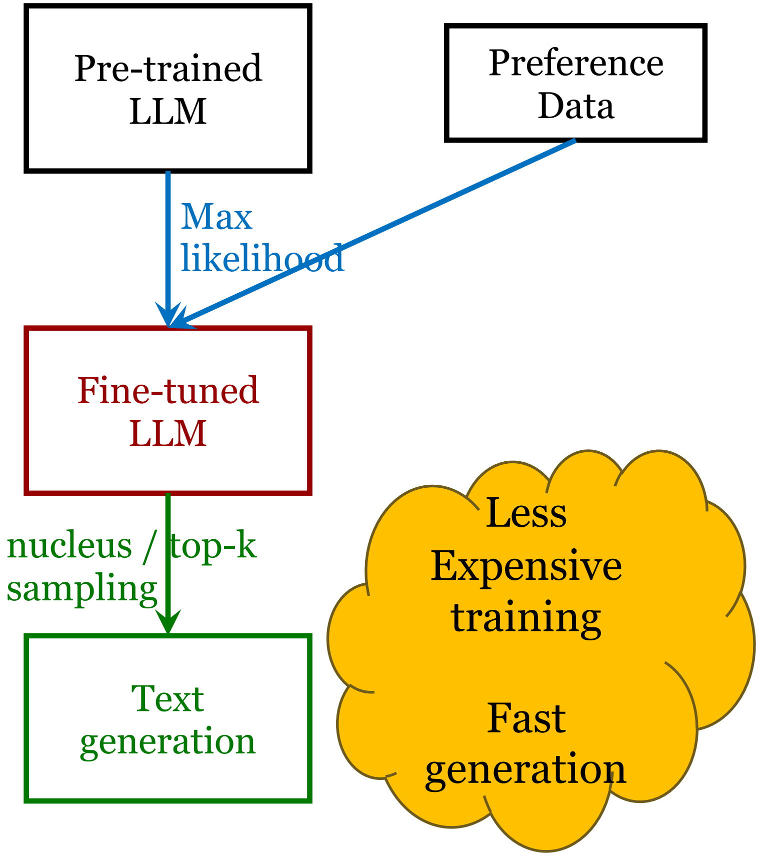
$$\max_{\phi} E_{D,\pi_\phi}[r_\phi(\boldsymbol{s}, \boldsymbol{a})] - \beta KL[\pi_\phi(\boldsymbol{a}|\boldsymbol{s})||\pi_{pretrained}(\boldsymbol{a}|\boldsymbol{s})]$$

RLHF Improvements

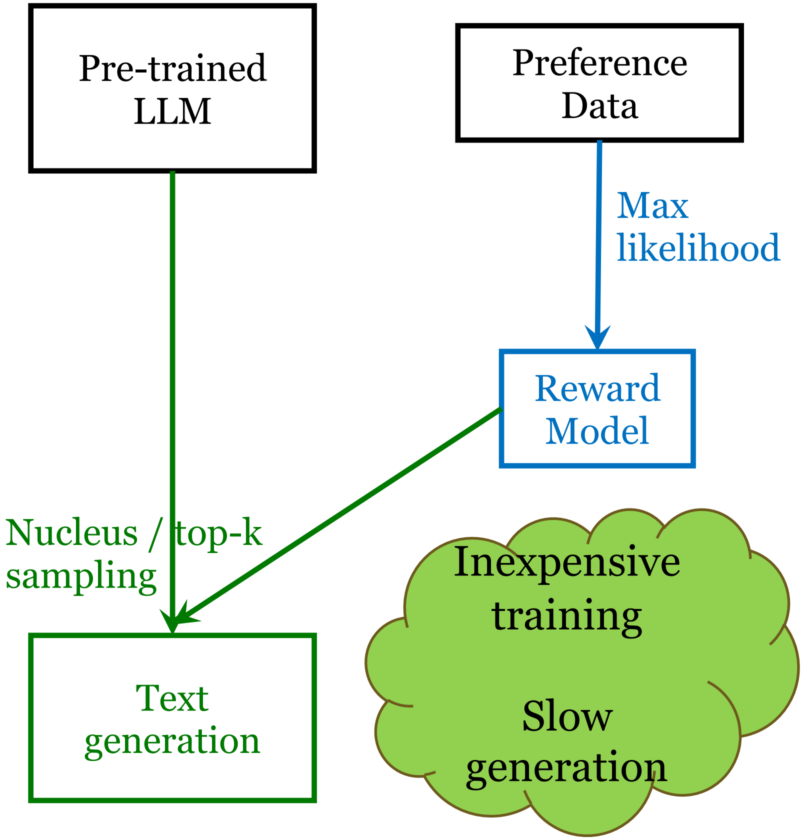
Proximal Policy Optimization (PPO)
Ouyang et al., 2022



Direct Preference Optimization (DPO)
Rafailov et al., 2023

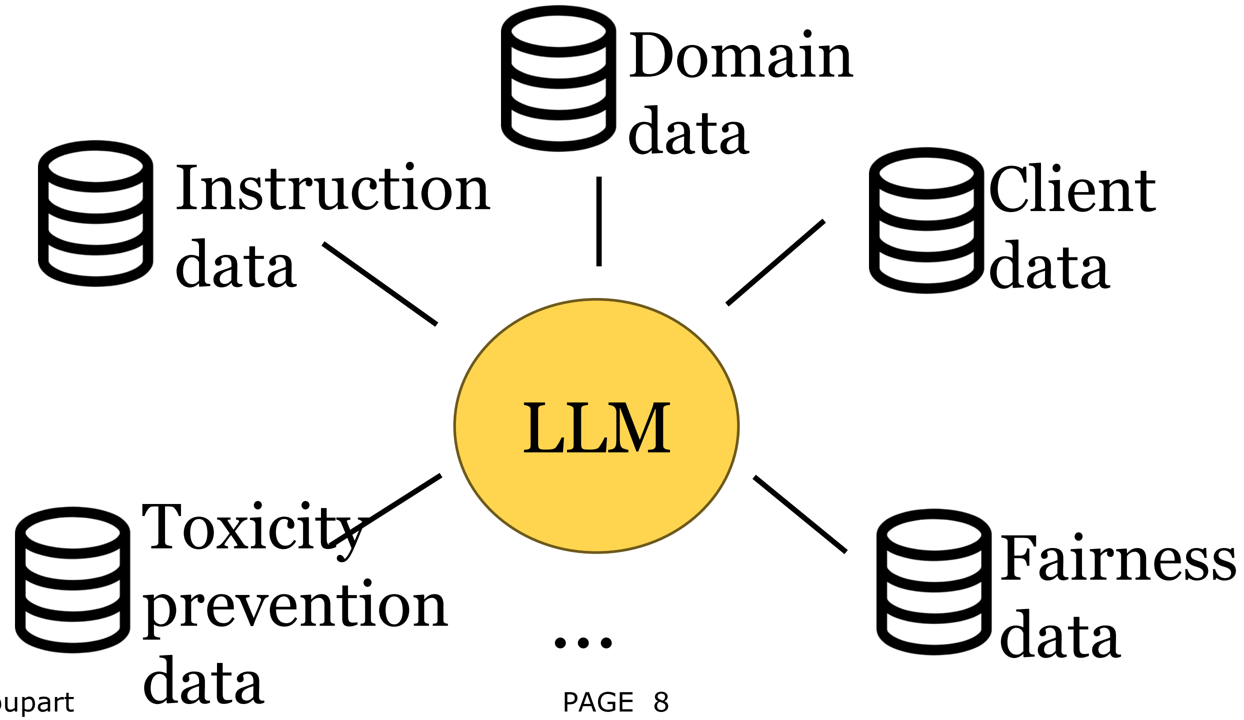


Reward Guided Text Generation (RGTG)
Khanov et al, 2024
Rashid et al., 2025



LLM Alignment with Preference Data

- Collect preference data: $D = \{(\boldsymbol{s}, \boldsymbol{a}_+, \boldsymbol{a}_-)_k\}_{k=1}^K$
  where $\boldsymbol{s}$: user prompt $\boldsymbol{a}$: system response
  $\boldsymbol{a}_+$ is preferred to $\boldsymbol{a}_-$ (i.e., $\boldsymbol{a}_+ \succ \boldsymbol{a}_-$)



Reward Model

Stiennon, Ouyang, Wu, Ziegler, Lowe Voss, Radford, Amodei, Christiano (2020) **Learning to summarize from human feedback**, *NeurIPS.*

- Reward function: $\boldsymbol{r_\theta}(\boldsymbol{s}, \boldsymbol{a}) = real\ number$
- Consider several possible responses $\boldsymbol{a}_1 \succcurlyeq \boldsymbol{a}_2 \succcurlyeq \cdots \succcurlyeq \boldsymbol{a}_k$ ranked by annotator
- Training reward function to be consistent with the ranking:

$$\boxed{Loss(\theta) = -\frac{1}{\binom{k}{2}} E_{(s,a_i,a_j)\in Dataset} \log \sigma\left(r_\theta(\boldsymbol{s}, \boldsymbol{a}_i) - r_\theta(\boldsymbol{s}, \boldsymbol{a}_j)\right)}$$

Reinforcement Learning

Ouyang, Wu, Jiang, Wainwright, et al. (2022) **Training language models to follow instructions with human feedback**, *NeurIPS*.

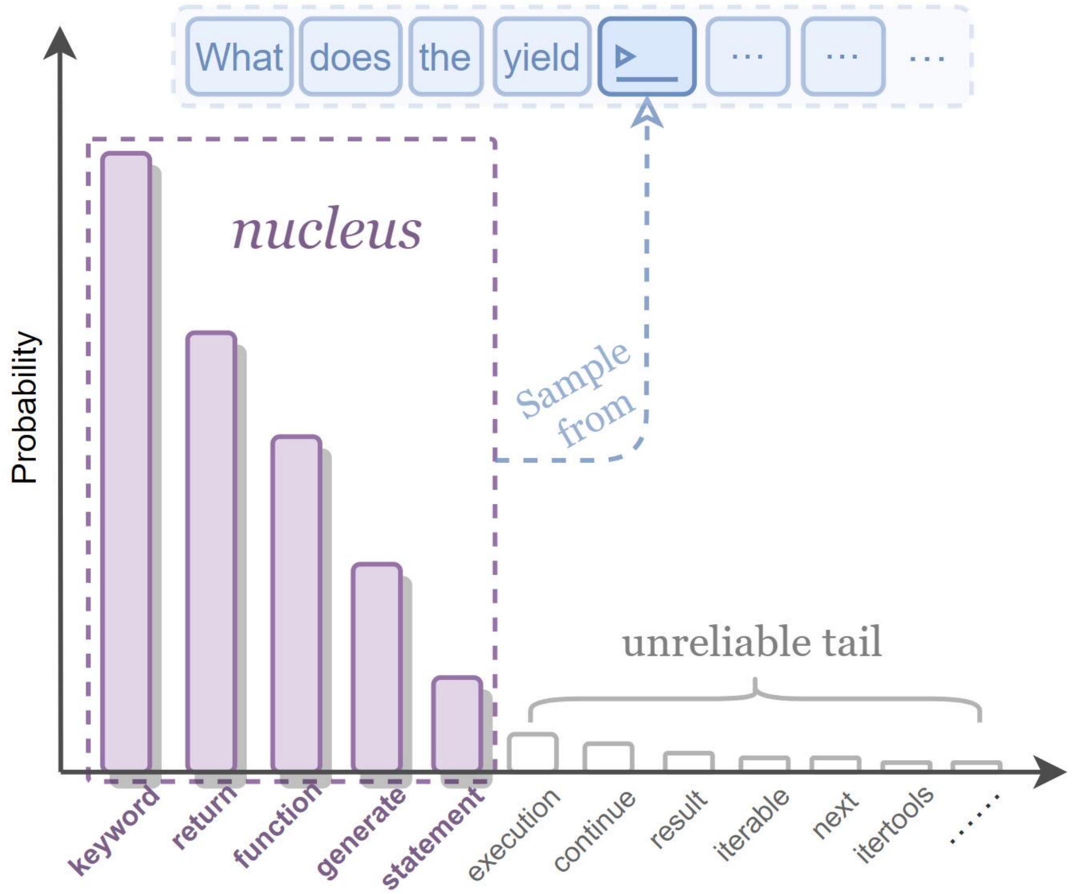
- Pretrain language model (GPT-3)
- Fine-Tune GPT-3 by RL to obtain InstructGPT
  - Policy (language model): $\pi_\phi(a|s)$
  - Optimize $\pi_\phi(s)$ by Proximal Policy Iteration (PPO)

$$\boxed{\max_\phi E_{s\in Dataset}\left[E_{a\sim\pi_\phi(a|s)}[r_\theta(s,a)] - \beta\, KL\big(\pi_\phi(\cdot\,|s)\big|\pi_{ref}(\cdot\,|s)\big)\right]}$$

Inference: Nucleus sampling

Sample from nucleus (top tokens only) to avoid unreliable responses while ensuring diversity

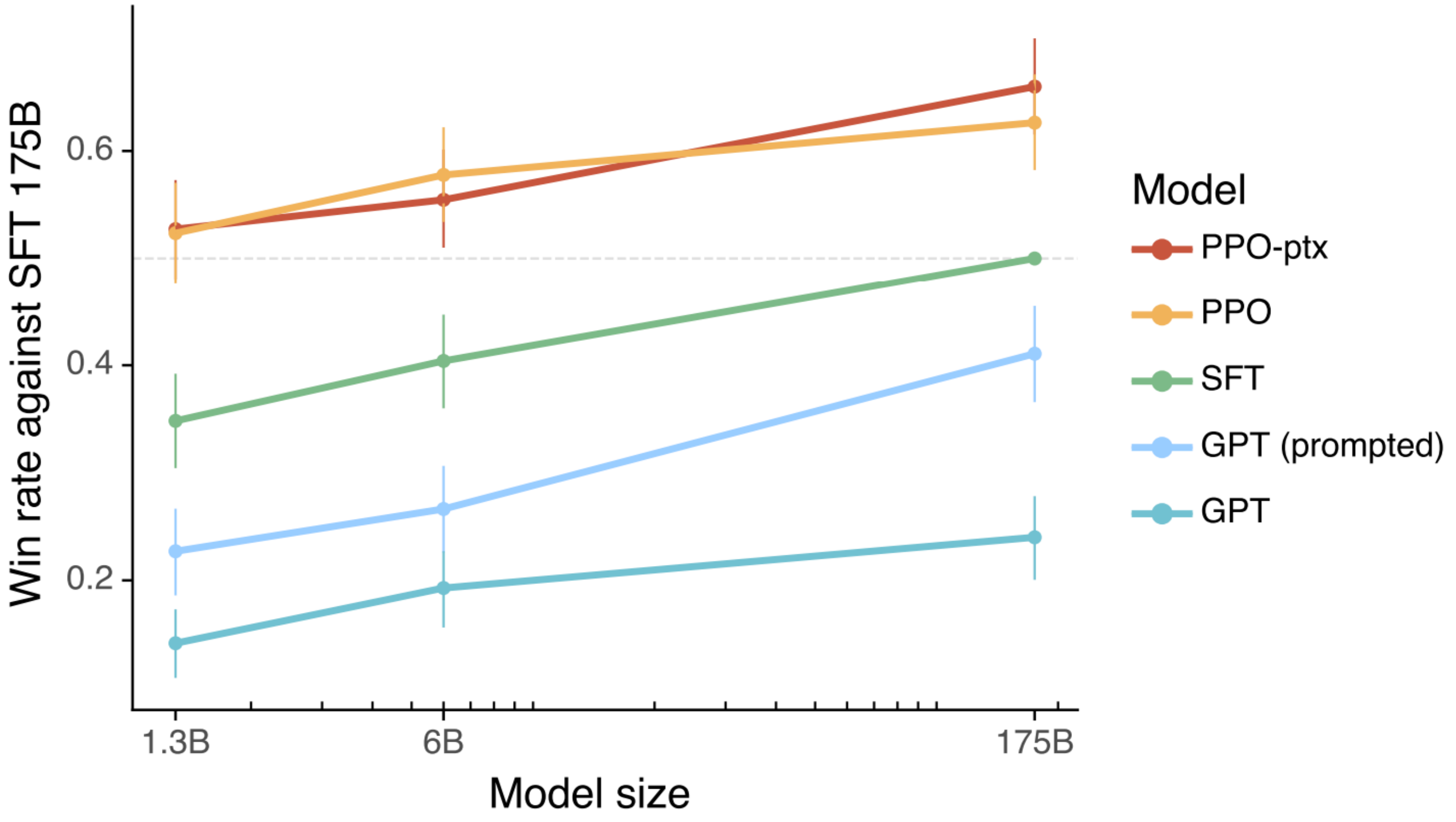
Holtzman, Ari; Buys, Jan; Du, Li; Forbes, Maxwell; Choi, Yejin (2019). **The Curious Case of Neural Text Degeneration**, arxiv.



Credit: https://ar5iv.labs.arxiv.org/html/2208.11523

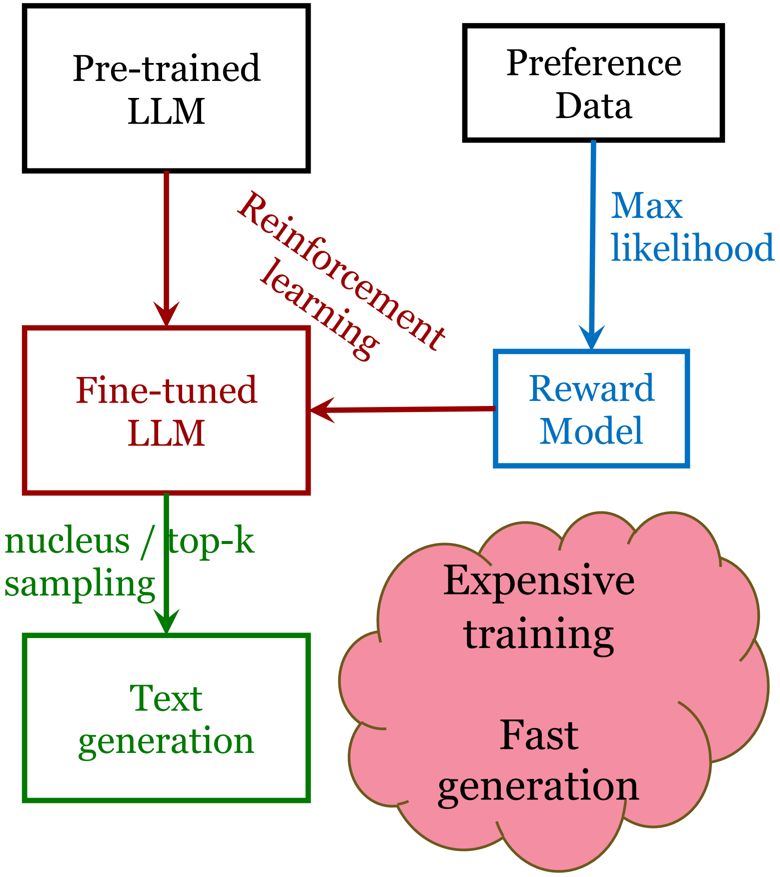
InstructGPT Results

Ouyang, Wu, Jiang, Wainwright, et al. (2022)

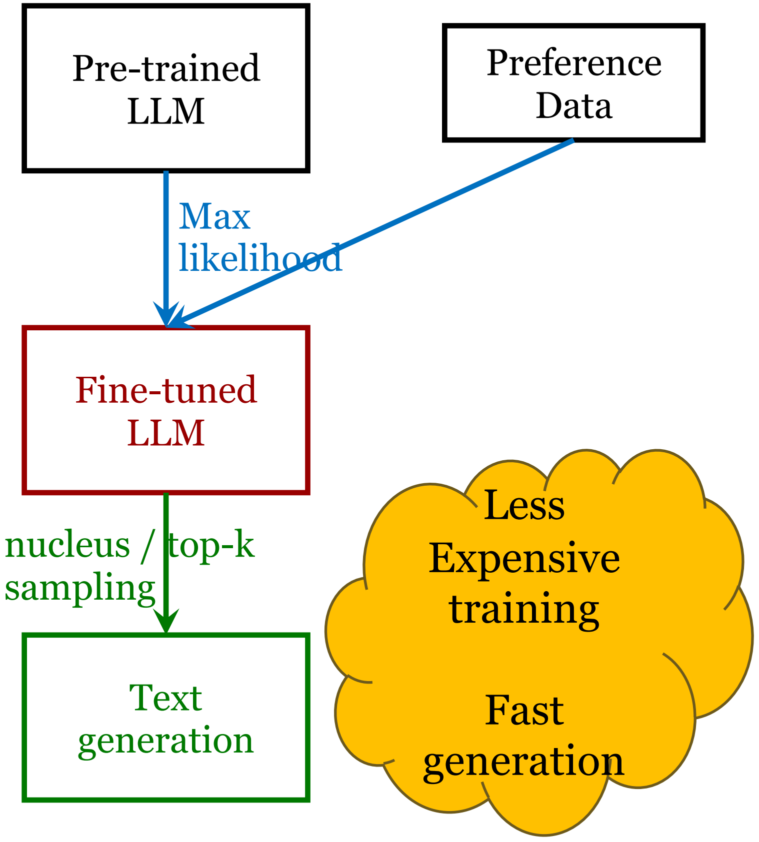


RLHF Improvements

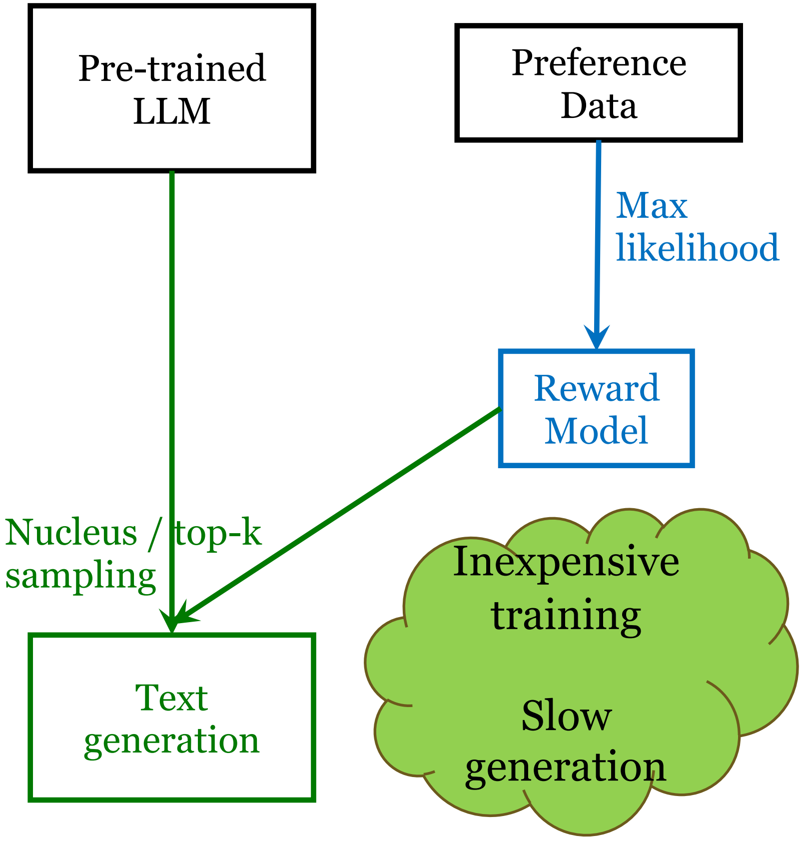
Proximal Policy Optimization (PPO)
Ouyang et al., 2022



Direct Preference Optimization (DPO)
Rafailov et al., 2023

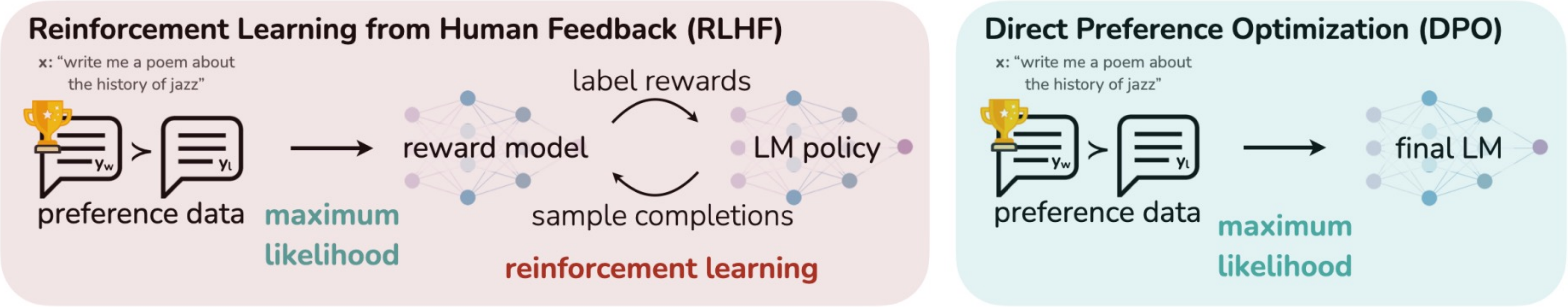


Reward Guided Text Generation (RGTG)
Khanov et al, 2024
Rashid et al., 2025



Direct Preference Optimization

Rafailov, Sharma, Mitchell, Ermon, Manning, Finn (2023) **Direct Preference Optimization: Your Language Model is Secretly a Reward Model**, *NeurIPS*.



Bypassing RL

- Recall RL objective:

$$\max_{\phi} E_{s \in Dataset}\left[E_{a \sim \pi_\phi(a|s)}[r_\theta(s,a)] - \beta\, KL\big(\pi_\phi(\cdot\,|s) \big| \pi_{ref}(\cdot\,|s)\big)\right]$$

- Closed form solution (based on maximum entropy RL):

$$\pi_\phi(a|s) = \frac{1}{Z(s)} \pi_{ref}(a|s) \exp\left(\frac{r_\theta(s,a)}{\beta}\right)$$

- Isolate reward: $r_\theta(s,a) = \beta \log \frac{\pi_\phi(a|s)}{\pi_{ref}(a|s)} + \beta \log Z(s)$
- Plug into preference objective:

$$Loss(\theta) = -\frac{1}{\binom{k}{2}} E_{(s,a_i,a_j) \in Dataset} \log \sigma\left(r_\theta(s,a_i) - r_\theta(s,a_j)\right)$$

$$= -\frac{1}{\binom{k}{2}} E_{(s,a_i,a_j) \in Dataset} \log \sigma\left(\beta \log \frac{\pi_\phi(a_i|s)}{\pi_{ref}(a_i|s)} - \beta \log \frac{\pi_\phi(a_j|s)}{\pi_{ref}(a_j|s)}\right)$$

Optimal Policy Derivation

$$\underset{\phi}{\operatorname{argmax}} E_{s\in Dataset}\left[E_{a\sim\pi_\phi(a|s)}[r_\theta(s,a)] - \beta\, KL\big(\pi_\phi(\cdot\,|s)\big|\pi_{ref}(\cdot\,|s)\big)\right]$$

$$= \underset{\phi}{\operatorname{argmax}} E_{s\in Dataset}\left[E_{a\sim\pi_\phi(a|s)}\left[r_\theta(s,a) - \beta\log\frac{\pi_\phi(a|s)}{\pi_{ref}(a|s)}\right]\right]$$ by KL definition

$$= \underset{\phi}{\operatorname{argmin}} E_{s\in Dataset}\left[E_{a\sim\pi_\phi(a|s)}\left[\log\frac{\pi_\phi(a|s)}{\pi_{ref}(a|s)} - \frac{1}{\beta}r_\theta(s,a)\right]\right]$$ since max = - min

$$= \underset{\phi}{\operatorname{argmin}} E_{s\in Dataset}\left[E_{a\sim\pi_\phi(a|s)}\left[\log\frac{\pi_\phi(a|s)}{\frac{1}{Z(s)}\pi_{ref}(a|s)\exp\left(\frac{r_\theta(s,a)}{\beta}\right)} - \log Z(s)\right]\right]$$ where $Z(s) = \sum_a \pi_{ref}(a|s)\exp\left(\frac{r_\theta(s,a)}{\beta}\right)$

$$= \underset{\phi}{\operatorname{argmin}} E_{s\in Dataset}\left[E_{a\sim\pi_\phi(a|s)}\left[\log\frac{\pi_\phi(a|s)}{\frac{1}{Z(s)}\pi_{ref}(a|s)\exp\left(\frac{r_\theta(s,a)}{\beta}\right)}\right]\right]$$ since $\log Z(s)$ is independent of ϕ

$$= \underset{\phi}{\operatorname{argmin}} E_{s\in Dataset}\left[E_{a\sim\pi_\phi(a|s)}\left[\log\frac{\pi_\phi(a|s)}{\pi_{\phi^*}(a|s)}\right]\right]$$ where $\pi_{\phi^*}(a|s) = \frac{1}{Z(s)}\pi_{ref}(a|s)\exp\left(\frac{r_\theta(s,a)}{\beta}\right)$

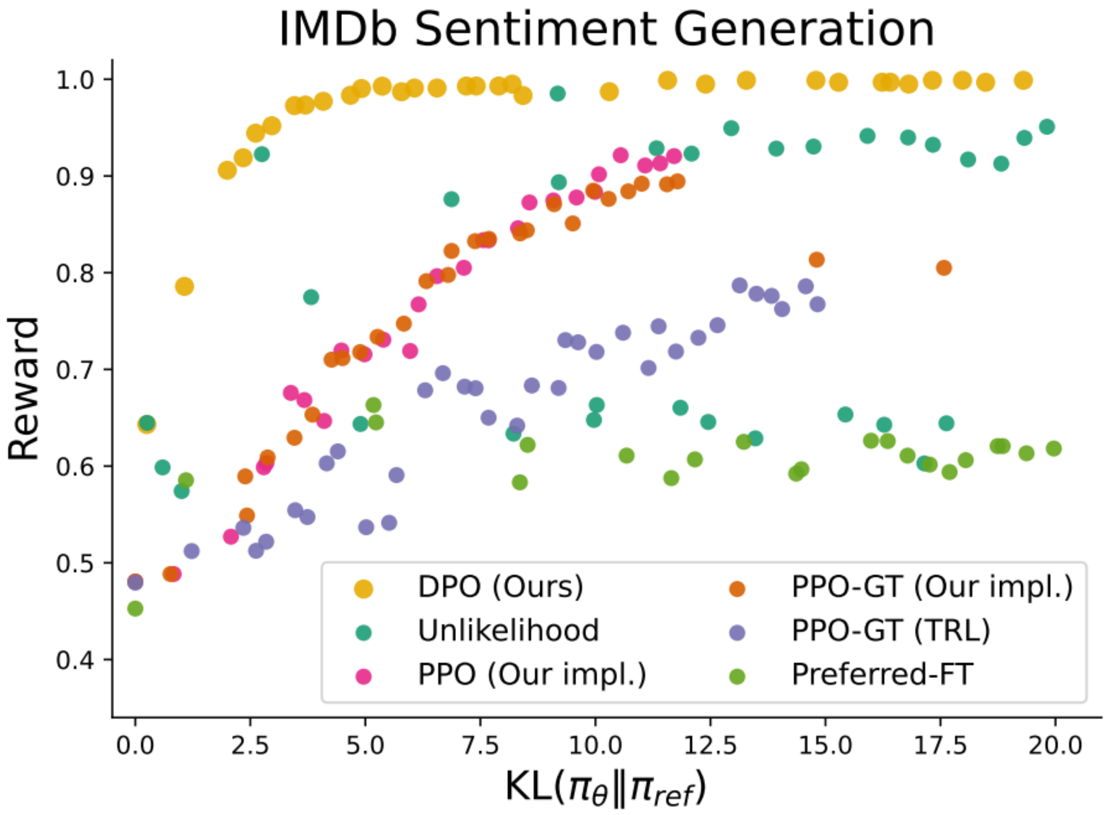
$$= \underset{\phi}{argmin}\, E_{s\in Dataset}\left[KL(\pi_\phi(\cdot\,|s)||\pi_{\phi^*}(\cdot\,|s))\right]$$ by KL definition

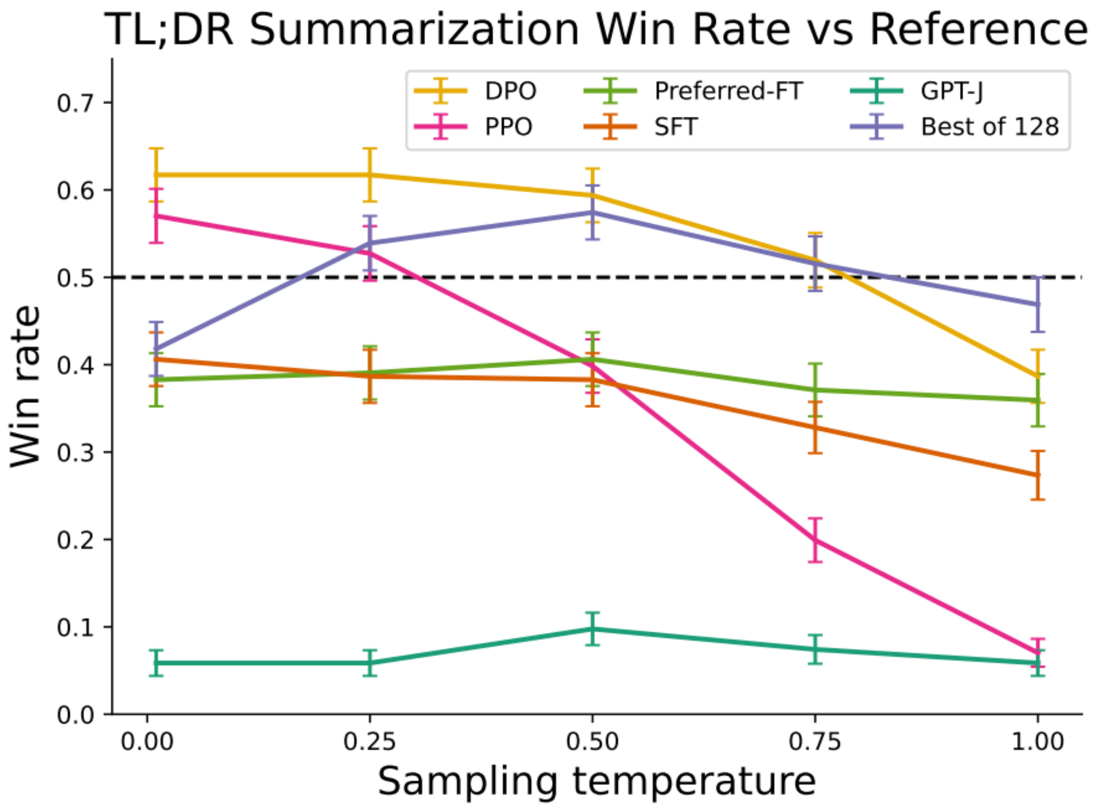
$$= \phi^*$$ since KL is minimized when both arguments are equal

UNIVERSITY OF WATERLOO

Empirical Results

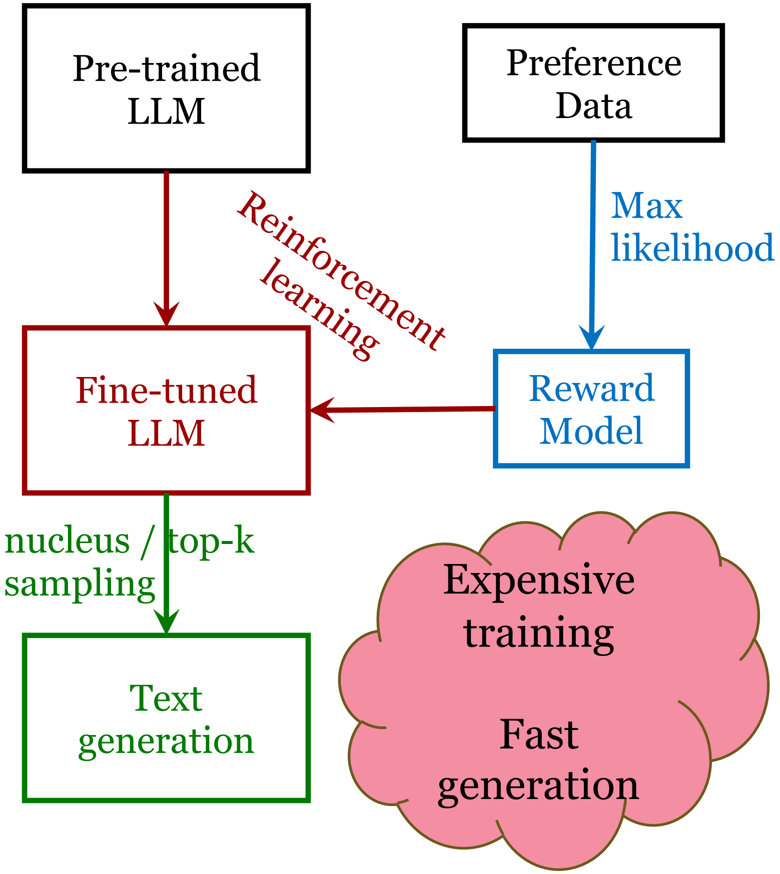
Rafailov et al. 2023



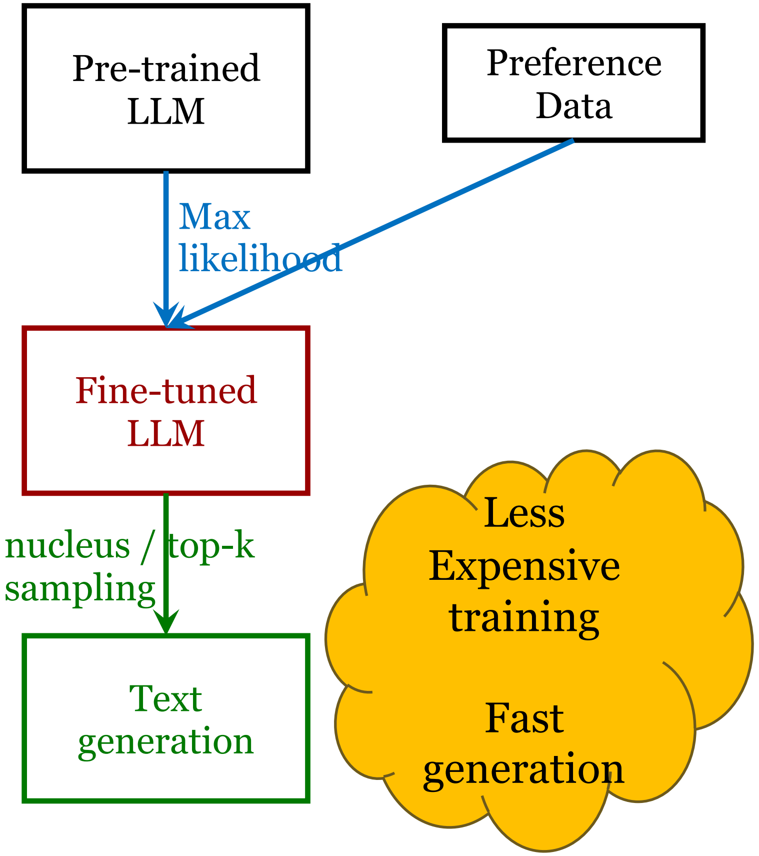


RLHF Improvements

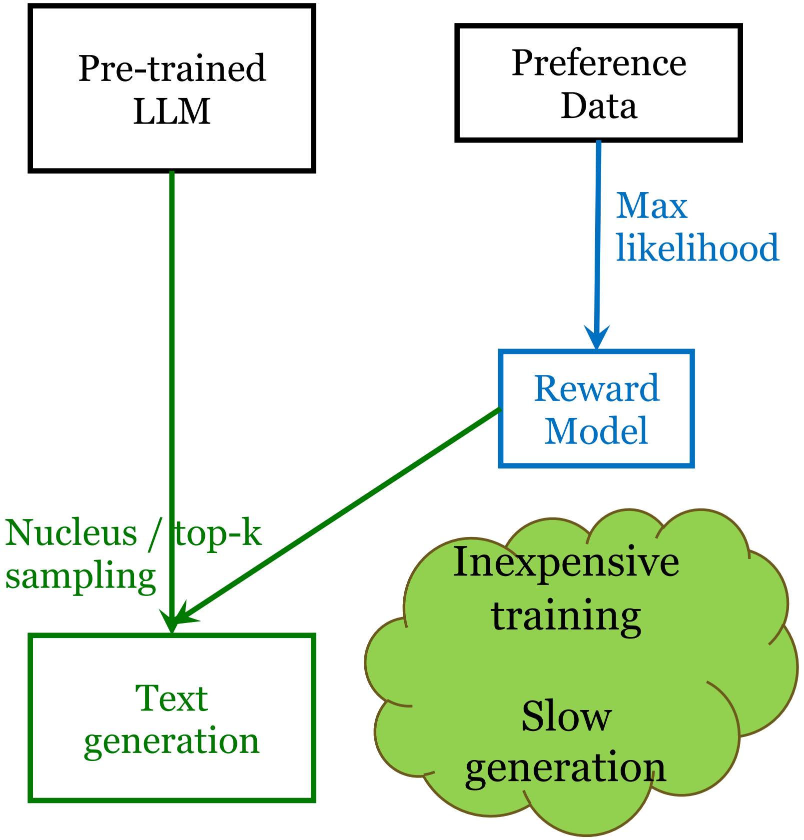
Proximal Policy Optimization (PPO)
Ouyang et al., 2022



Direct Preference Optimization (DPO)
Rafailov et al., 2023



Reward Guided Text Generation (RGTG)
Khanov et al, 2024
Rashid et al., 2025



UNIVERSITY OF WATERLOO

Sequence Generation

- Recall closed form solution

$$\pi_\phi(\boldsymbol{a}|\boldsymbol{s}) = \frac{1}{Z(\boldsymbol{s})}\pi_{ref}(\boldsymbol{a}|\boldsymbol{s})\exp\left(\frac{r_\theta(\boldsymbol{s},\boldsymbol{a})}{\beta}\right)$$

$$= softmax\left(\log\pi_{ref}(\boldsymbol{a}|\boldsymbol{s}) + \frac{r_\theta(\boldsymbol{s},\boldsymbol{a})}{\beta}\right)$$

- Text generation:

$$\boldsymbol{a} \sim softmax\left(\log\begin{pmatrix}\pi_{ref}(\boldsymbol{a}_1|\boldsymbol{s})\\ \pi_{ref}(\boldsymbol{a}_2|\boldsymbol{s})\\ \pi_{ref}(\boldsymbol{a}_3|\boldsymbol{s})\\ \dots\\ \pi_{ref}(\boldsymbol{a}_n|\boldsymbol{s})\end{pmatrix} + \begin{pmatrix}r_\theta(\boldsymbol{s},\boldsymbol{a}_1)\\ r_\theta(\boldsymbol{s},\boldsymbol{a}_2)\\ r_\theta(\boldsymbol{s},\boldsymbol{a}_3)\\ \dots\\ r_\theta(\boldsymbol{s},\boldsymbol{a}_n)\end{pmatrix}/\beta\right)$$

Token Generation

- Token-wise LLM modeling

$$\pi_\phi(a^i|\boldsymbol{s},\boldsymbol{a}^{1:i-1}) = \frac{1}{Z(\boldsymbol{s})}\pi_{ref}(a^i|\boldsymbol{s},\boldsymbol{a}^{1:i-1})\exp\left(\frac{r_\theta(\boldsymbol{s},\boldsymbol{a}^{1:i})}{\beta}\right)$$

$$= softmax\left(\log \pi_{ref}(a^i|\boldsymbol{s},\boldsymbol{a}^{1:i-1}) + \frac{r_\theta(\boldsymbol{s},\boldsymbol{a}^{1:i})}{\beta}\right)$$

- Token generation:

$$a^i \sim softmax\left(\log\begin{pmatrix}\pi_{ref}(a_1^i|\boldsymbol{s},\boldsymbol{a}^{1:i-1})\\ \pi_{ref}(a_2^i|\boldsymbol{s},\boldsymbol{a}^{1:i-1})\\ \pi_{ref}(a_3^i|\boldsymbol{s},\boldsymbol{a}^{1:i-1})\\ \dots\\ \pi_{ref}(a_n^i|\boldsymbol{s},\boldsymbol{a}^{1:i-1})\end{pmatrix} + \begin{pmatrix}r_\theta(\boldsymbol{s},\boldsymbol{a}^{1:i-1},a_1^i)\\ r_\theta(\boldsymbol{s},\boldsymbol{a}^{1:i-1},a_2^i)\\ r_\theta(\boldsymbol{s},\boldsymbol{a}^{1:i-1},a_3^i)\\ \dots\\ r_\theta(\boldsymbol{s},\boldsymbol{a}^{1:i-1},a_n^i)\end{pmatrix}/\beta\right)$$

FaRMA: Faster Reward Model for Alignment

- Rashid, Wu, Fan, Li, Kristiadi, Poupart (2025) **Towards Cost-Effective Reward Guided Text Generation**, arxiv.
- Optimization problem:

$$\max_{\theta} E_{(\boldsymbol{s},\boldsymbol{a}_+,\boldsymbol{a}_-)\in Dataset} \log \sigma\left(r_\theta(\boldsymbol{s},\boldsymbol{a}_+) - r_\theta(\boldsymbol{s},\boldsymbol{a}_-)\right)$$
$$\text{Subject to } r_\theta\left(\boldsymbol{s},\boldsymbol{a}^{1:i}\right) = \max_{\boldsymbol{a}^{i+1:|\boldsymbol{a}|}} r_\theta\left(\boldsymbol{s},[\boldsymbol{a}^{1:i},\boldsymbol{a}^{i+1:|\boldsymbol{a}|}]\right)\ \forall \boldsymbol{s},\boldsymbol{a},i$$

- In practice: alternate between minimizing two loss functions
  - $L_1(\theta) = -E_{(\boldsymbol{s},\boldsymbol{a}_+,\boldsymbol{a}_-)\in Dataset} \log \sigma\left(r_\theta(\boldsymbol{s},\boldsymbol{a}_+) - r_\theta(\boldsymbol{s},\boldsymbol{a}_-)\right)$
  - $L_2(\theta) = \frac{1}{2} E_{(\boldsymbol{s},\boldsymbol{a})\in Dataset, i\le|\boldsymbol{a}|}\left(r_\theta(\boldsymbol{s},\boldsymbol{a}^{1:i}) - \max_{\boldsymbol{a}^{i+1:|\boldsymbol{a}|}} r_\theta\left(\boldsymbol{s},[\boldsymbol{a}^{1:i},\boldsymbol{a}^{i+1:|\boldsymbol{a}|}]\right)\right)^2$

FaRMA Pseudocode

Repeat

  Repeat for each $(s, \boldsymbol{a}_+, \boldsymbol{a}_-)$ in minibatch

$$L_1(\theta) = \log \sigma\left(r_\theta(\boldsymbol{s}, \boldsymbol{a}_+) - r_\theta(\boldsymbol{s}, \boldsymbol{a}_-)\right)$$

$$\theta \leftarrow \theta \; - \alpha \nabla L_1(\theta)$$

  Repeat for each $(\boldsymbol{s}, \boldsymbol{a}, i)$ in minibatch

$$L_2(\theta) = \frac{1}{2}\left(r_\theta\left(\boldsymbol{s}, \boldsymbol{a}^{1:i}\right) - \max_{a^{i+1}} r_\theta\left(\boldsymbol{s}, \boldsymbol{a}^{1:i+1}\right)\right)^2$$

$$\theta \leftarrow \theta \; - \alpha \nabla L_2(\theta)$$

UNIVERSITY OF WATERLOO

Empirical Results

TL;DR Summarization			
Method	**LLM**	$r \pm$ SE	**Time(min)**
π_{ref}	frozen	0.98±0.18	2
ARGS	frozen	1.46±0.16	32
PARGS	frozen	1.56±0.19	31
CD	frozen	1.15±0.16	29
FaRMA	frozen	2.05±0.15	5
CARDS	frozen	1.73±0.16	17
DPO	trained	2.08±0.18	2
PPO	trained	2.05±0.14	2

Table 2. Avg. reward (over 100 samples) ± standard error total generation time for the TL;DR summarization task.

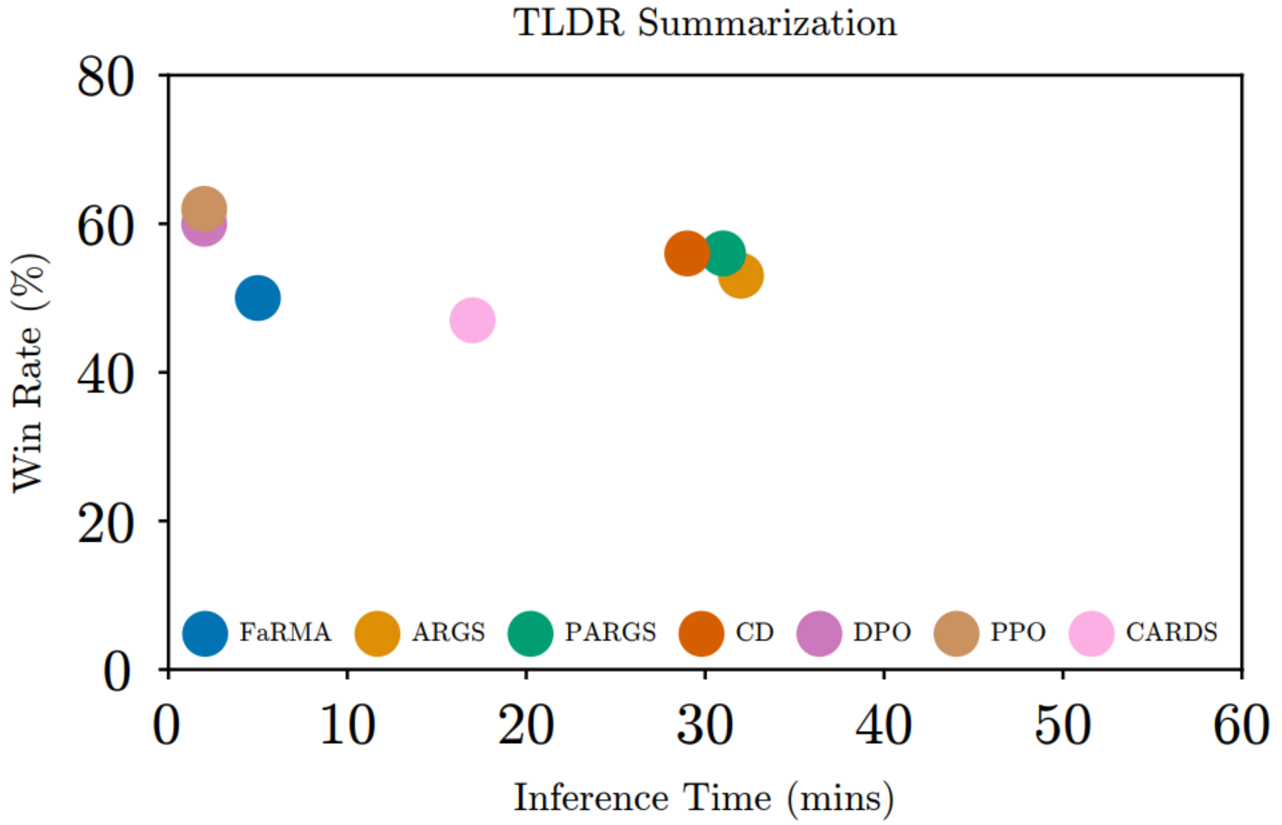


Figure 2. GPT4 evaluation on TLDR

Towards Plug-n-play LLMs

Large language models Preference Datasets