

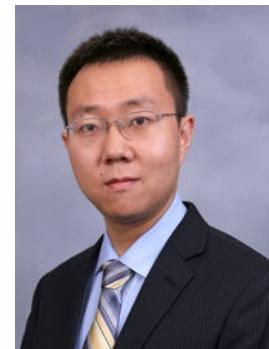
# ComPO: Preference Alignment via Comparison Oracles

Peter Chen

Department of Mathematics,  
Columbia University

Atlanta  
Oct, 2025

Advances in Neural Information Processing Systems 38,  
joint work with

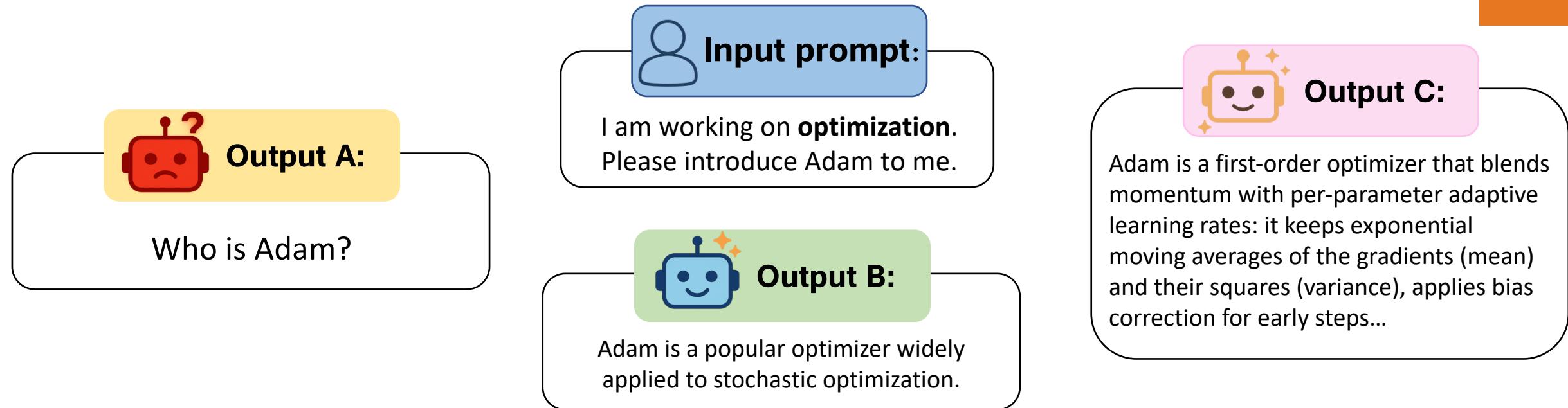


**Xi Chen**  
NYU Stern

**Wotao Yin**  
Alibaba U.S.

**Tianyi Lin**  
Columbia

# LLM Alignment from Human Preference Feedback

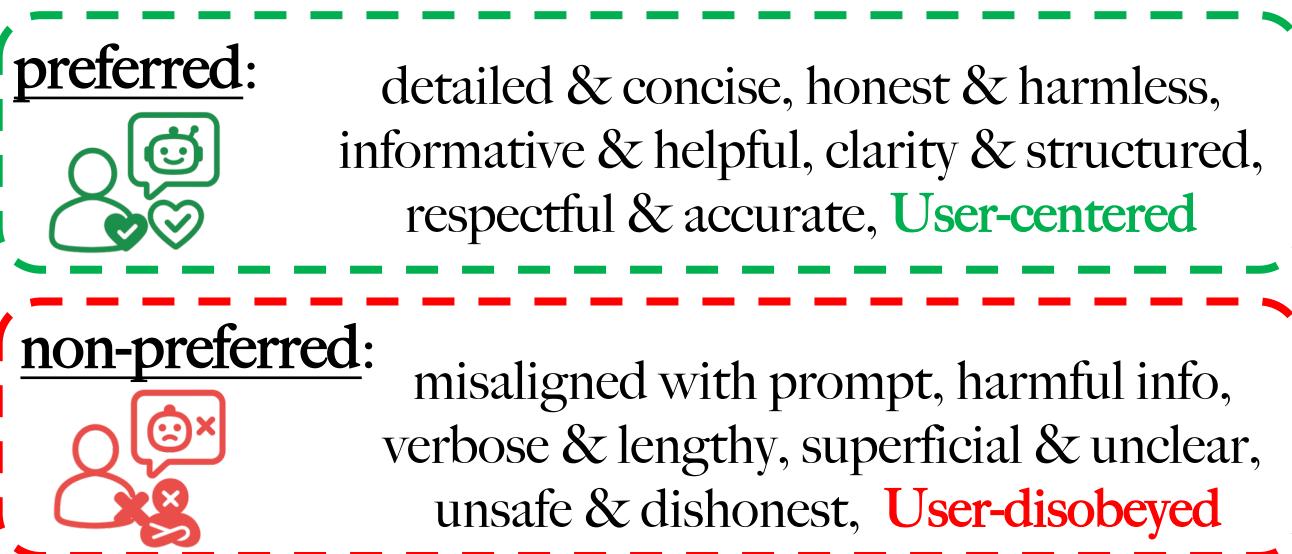


Human Preference Feedback: C > B > A

A - Factual mistake

B - Correct, but too higher-level

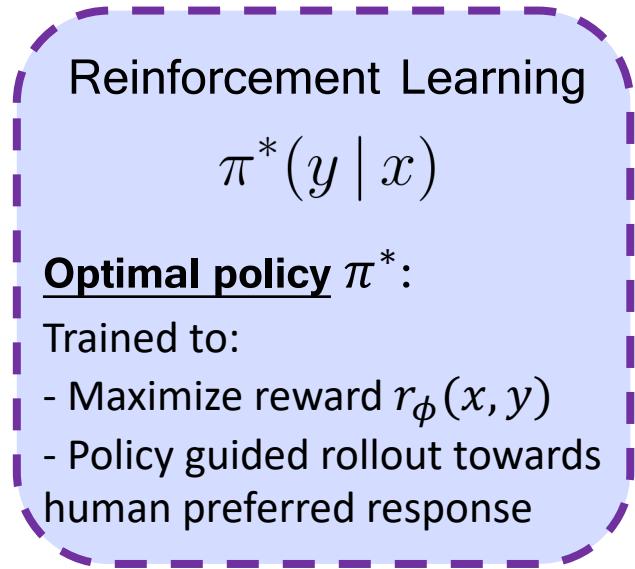
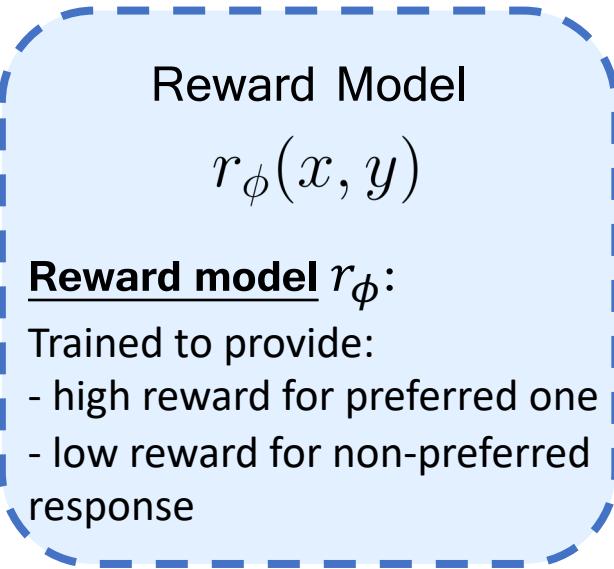
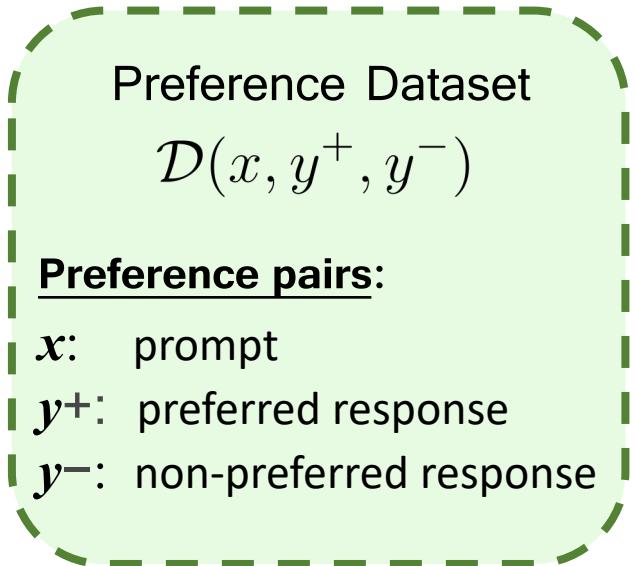
C - Correct, detailed and informative



# RLHF: Reinforcement Learning from Human Feedback



Goal: encourage preferred output & discourage non-preferred output



## Obstacles:

- (i) **Resource-intensive** online training: reward & policy model training
- (ii) Heuristics raised from training due to **inaccurate reward provision**

Can we reduce it into an offline training, with a more deterministic objective/policy update?

# DPO: Direct Preference Optimization



NEURAL INFORMATION  
PROCESSING SYSTEMS

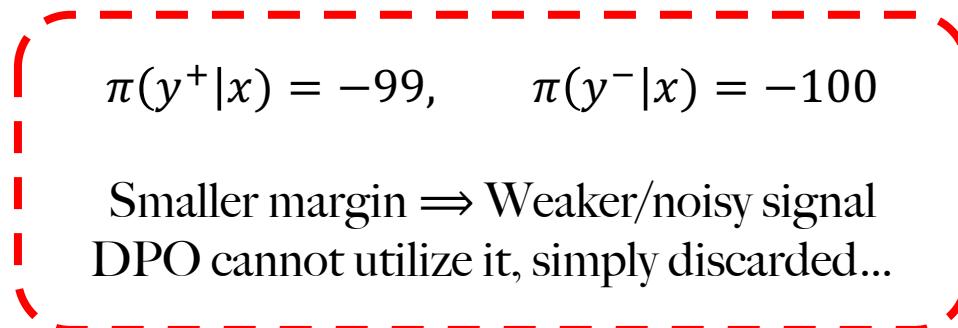
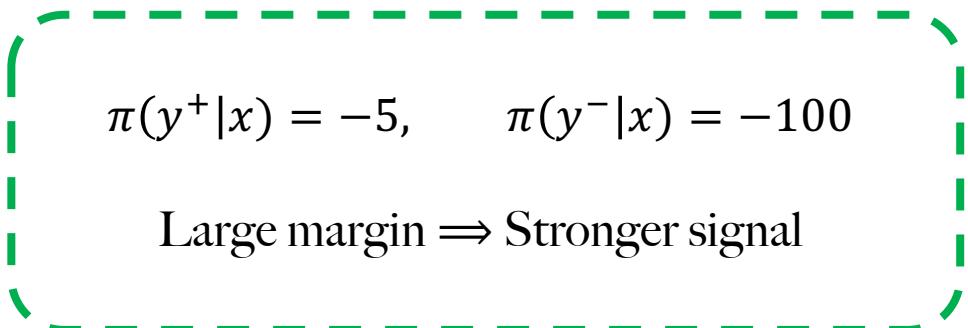
informs  
ANNUAL  
MEETING

DPO loss: a reparameterization from RLHF objective

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = \mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[ \log \sigma \left( \beta \log \frac{\pi_\theta(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \beta \log \frac{\pi_\theta(y^-|x)}{\pi_{\text{ref}}(y^-|x)} \right) \right]$$

**Likelihood margin:**

We want larger margin between each winning/losing pair



Can we extract useful information from these noisy preference pairs?

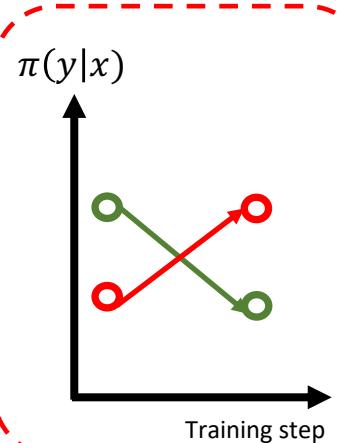
# ComPO: Zeroth-order, Comparison-based method

DPO: Extract preference info via external **margin difference**

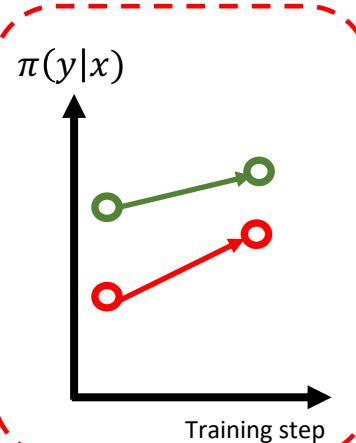
⇒ failed when such difference is numerically small, especially under noisy pairs

ComPO: Extract preference info through internal **contrastive relation**

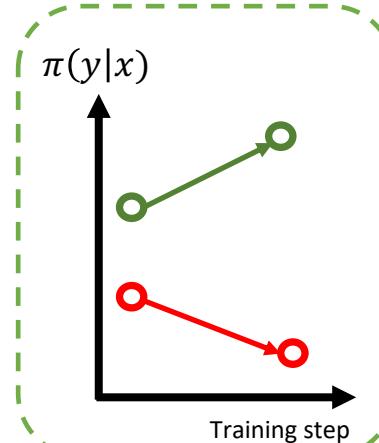
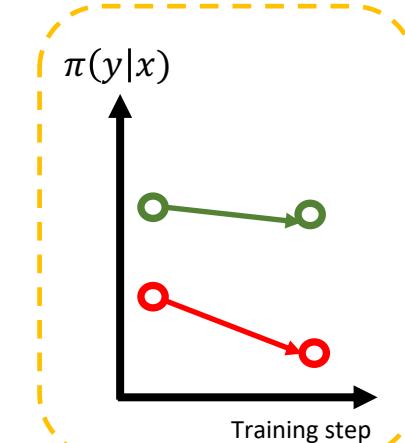
- Preferred response
- Non-preferred response



(a) Misalignment



(b) Likelihood displacement  
(Razin et al., 2025)



Expected

Given that  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is a function where neither its function value nor its gradient is accessible, we define a pairwise comparison oracle  $\mathcal{C}_f$  in its simplest form as follows,

**Definition 2.1** We call  $\mathcal{C}_f(\theta, \theta') : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \{+1, -1\}$  a comparison oracle for function  $f$  if

$$\mathcal{C}_f(\theta, \theta') = \begin{cases} -1, & \text{if } f(\theta') < f(\theta), \\ +1, & \text{otherwise.} \end{cases}$$

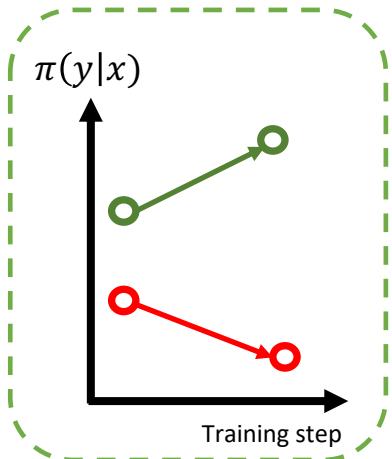
Goal: minimizing  $f$ , but only get access to implicit comparison between two sets of model parameter  $\theta$

tl;dr “-1 returned:  $\theta'$  is better than  $\theta$  for  $f$ ”

# Preference Comparison Oracles

**Definition 3.1** We say  $\mathcal{C}_\pi(\theta, \theta') : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \{+1, -1\}$  a preference comparison oracle for the model  $\pi_\theta$  and a pair of preference data  $(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-)$  from the offline dataset  $\mathcal{D}$  if

$$\mathcal{C}_\pi(\theta, \theta') = \begin{cases} -1, & \text{if } \pi_{\theta'}(\mathbf{y}^+|\mathbf{x}) > \pi_\theta(\mathbf{y}^+|\mathbf{x}) \text{ and } \pi_{\theta'}(\mathbf{y}^-|\mathbf{x}) < \pi_\theta(\mathbf{y}^-|\mathbf{x}) \text{ for } (\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-) \in \mathcal{D}, \\ +1, & \text{otherwise.} \end{cases}$$



One-bit Compressed Approximation (Cai et al., 2022):

$$\mathcal{C}_f(\theta, \theta + r\mathbf{z}_i) \approx \text{sign}(f(\theta + r\mathbf{z}_i) - f(\theta)) \approx \text{sign}(\mathbf{z}_i^\top \nabla f(\theta))$$

↓ Random perturbation  $z$  within radius  $r$

Gradient with Sparse Structure: Define oracle feedback  $y_i = \mathcal{C}_f(\theta, \theta + r\mathbf{z}_i)$ , we have gradient

$$\hat{\mathbf{g}} = \underset{\substack{\|\mathbf{g}\|_1 \leq \sqrt{s} \\ \|\mathbf{g}\| \leq 1}}{\operatorname{argmax}} \sum_{i=1}^m y_i \mathbf{z}_i^\top \mathbf{g}$$

Choose  $\mathbf{g}$  to align with the direction from random perturbed vectors & oracle feedback; think as a dot product

Zeroth-order gives noisy gradient  
make it sparse to reduce noise

# Billion-level Parameter Space

$$\hat{\mathbf{g}} = \operatorname{argmax}_{\|\mathbf{g}\|_1 \leq \sqrt{s}, \|\mathbf{g}\| \leq 1} \sum_{i=1}^m y_i \mathbf{z}_i^\top \mathbf{g}$$

$$\hat{\mathbf{g}} = \operatorname{argmax}_{\|\mathbf{g}\|_1 \leq \sqrt{s}, \|\mathbf{g}\| \leq 1} \sum_{i=1}^m y_i \mathbf{z}_i^\top \mathbf{g}$$

$$\hat{\mathbf{g}}^o = \frac{\sum_{i=1}^m y_i \mathbf{z}_i}{\|\sum_{i=1}^m y_i \mathbf{z}_i\|}$$

Prior works solved within the dimensionality of 500,  
but what about LLMs, with **billion** parameters?

Step 1: Sparsity simplification  
Temporarily **eliminate** the sparsity norm constraint

Step 2: Derive simplified gradient  
Retrieve with closed form

Step 3: Apply gradient-entry threshold  $\lambda_g$   
Clip noisy gradient entry with lower magnitude, bringing sparsity

Step 4: Dynamic stepwise adjustment + threshold  
Filtering out gradient with less successful oracle feedbacks

---

## Algorithm 2 Comparison-Based Preference Alignment (Practical Scheme)

---

- 1: **Input:** initial parameter  $\theta_1 = [\bar{\theta}_1; \theta_t^o] \in \mathbb{R}^d$ , scaling for stepsize  $\gamma > 0$ , sampling radius  $r > 0$ , querying number  $m \geq 1$ , clipping thresholds  $\lambda_g, \lambda > 0$ , and iteration number  $T \geq 1$ .
- 2: **for**  $t = 1, 2, \dots, T$  **do**
- 3:     Draw  $m$  i.i.d. samples uniformly from a unit sphere in  $\mathbb{R}^{d^o}$ , i.e.,  $\{\mathbf{z}_i\}_{1 \leq i \leq m}$ .
- 4:     Compute  $y_i = \mathcal{C}_\pi([\bar{\theta}_1; \theta_t^o], [\bar{\theta}_1; \theta_t^o + r\mathbf{z}_i])$  for  $i = 1, 2, \dots, m$ .
- 5:     Compute  $\hat{\mathbf{g}}_t^o = \frac{\sum_{i=1}^m y_i \mathbf{z}_i}{\|\sum_{i=1}^m y_i \mathbf{z}_i\|}$  and clip  $\hat{\mathbf{g}}_t^o$  by zeroing out the entries whose magnitude is less than  $\boxed{\lambda_g}$ .
- 6:     Compute  $\theta_{t+1}^o = \theta_t^o - \frac{\gamma |\{i: y_i = -1\}|}{m} \hat{\mathbf{g}}_t^o$  if  $\frac{|\{i: y_i = -1\}|}{m} > \boxed{\lambda}$  and  $\theta_{t+1}^o = \theta_t^o$  otherwise.



*Gradient entry threshold*

*Stepsize threshold (oracle feedback)*

ComPO is trained on the noisy pairs that DPO cannot effectively utilize, i.e., with

$$\|\log \pi_{\text{ref}}(\mathbf{y}^+ | \mathbf{x}) - \log \pi_{\text{ref}}(\mathbf{y}^- | \mathbf{x})\| \leq \delta,$$

\*  $\delta = 3$  is used in actual implementation, with running ComPO on the first 100 noisy pairs

# Directly augmenting DPO

Table 1: Evaluation results on AlpacaEval 2, Arena-Hard, and MT-Bench under four model setups. LC and WR denote length-controlled win rate and win rate, respectively. Turn-1 and Turn-2 represent the scores to the answers from the first and follow-up questions in multi-turn dialogue. Here, we run 5 trials for DPO<sub>clean</sub>+ComPO and present the best trial performance.

| Method                      | Mistral-Base-7B |             |            |  |             |             | Mistral-Instruct-7B |              |              |             |             |             |             |
|-----------------------------|-----------------|-------------|------------|--|-------------|-------------|---------------------|--------------|--------------|-------------|-------------|-------------|-------------|
|                             | AlpacaEval 2    |             | Arena-Hard |  | MT-Bench    |             | AlpacaEval 2        |              | Arena-Hard   |             | MT-Bench    |             |             |
|                             | LC (%)          | WR (%)      | WR (%)     |  | Turn-1      | Turn-2      | Avg.                | LC (%)       | WR (%)       | WR (%)      | Turn-1      | Turn-2      | Avg.        |
| DPO                         | 9.71            | 6.27        | 2.9        |  | 6.20        | <b>5.38</b> | <b>5.79</b>         | 24.14        | 16.71        | <b>14.4</b> | 6.28        | 5.42        | 5.86        |
| DPO <sub>clean</sub>        | 9.41            | 6.52        | 3.0        |  | 6.18        | 5.22        | 5.70                | 23.89        | 16.15        | 14.2        | 6.11        | 5.34        | 5.73        |
| DPO <sub>clean</sub> +ComPO | <b>11.66</b>    | <b>6.55</b> | <b>3.2</b> |  | <b>6.22</b> | 5.32        | 5.77                | <b>26.17</b> | <b>18.32</b> | 10.5        | <b>7.78</b> | <b>7.63</b> | <b>7.69</b> |

| Method                      | Llama-3-Base-8B |              |             |  |             |             | Llama-3-Instruct-8B |              |              |             |             |             |             |
|-----------------------------|-----------------|--------------|-------------|--|-------------|-------------|---------------------|--------------|--------------|-------------|-------------|-------------|-------------|
|                             | AlpacaEval 2    |              | Arena-Hard  |  | MT-Bench    |             | AlpacaEval 2        |              | Arena-Hard   |             | MT-Bench    |             |             |
|                             | LC (%)          | WR (%)       | WR (%)      |  | Turn-1      | Turn-2      | Avg.                | LC (%)       | WR (%)       | WR (%)      | Turn-1      | Turn-2      | Avg.        |
| DPO                         | 4.14            | 10.43        | <b>12.1</b> |  | 6.61        | 5.85        | 6.23                | 32.59        | 31.99        | 22.9        | 8.30        | 7.55        | 7.93        |
| DPO <sub>clean</sub>        | 4.28            | 9.81         | 12.0        |  | <b>6.64</b> | 6.01        | 6.33                | 32.92        | 32.42        | 22.9        | 8.26        | 7.63        | 7.94        |
| DPO <sub>clean</sub> +ComPO | <b>5.39</b>     | <b>10.93</b> | <b>12.1</b> |  | 6.60        | <b>6.28</b> | <b>6.44</b>         | <b>35.79</b> | <b>35.03</b> | <b>23.1</b> | <b>8.39</b> | <b>7.71</b> | <b>8.05</b> |

LC: Length-controlled winning rate against base model (GPT 4-Turbo), judged by GPT 4

WR: Winning rate (without length control) against base model (GPT 4-Turbo), judged by GPT 4

DPO<sub>clean</sub>: Running DPO while excluding those noisy pairs

Table 2: Direct augmentation results on SimPO over different models and benchmarks.

| <b>Model</b>               | <b>Method</b> | <b>AlpacaEval 2</b> |              | <b>Arena-Hard</b> | <b>MT-Bench</b> |             |             |
|----------------------------|---------------|---------------------|--------------|-------------------|-----------------|-------------|-------------|
|                            |               | LC (%)              | WR (%)       | WR (%)            | Turn-1          | Turn-2      | Avg.        |
| <b>Mistral-Instruct-7B</b> | SimPO         | 40.22               | 41.18        | 20.8              | <b>7.94</b>     | 7.31        | 7.62        |
|                            | SimPO + ComPO | <b>42.27</b>        | <b>43.17</b> | <b>22.0</b>       | 7.83            | <b>7.46</b> | <b>7.64</b> |
| <b>Llama-3-Instruct-8B</b> | SimPO         | 48.71               | 43.66        | 36.3              | 7.91            | 7.42        | 7.66        |
|                            | SimPO + ComPO | <b>49.53</b>        | <b>45.03</b> | <b>37.3</b>       | <b>7.94</b>     | <b>7.45</b> | <b>7.70</b> |
| <b>Gemma-2-it-9B</b>       | SimPO         | 60.36               | 55.59        | <b>61.1</b>       | <b>9.07</b>     | 8.47        | 8.77        |
|                            | SimPO + ComPO | <b>62.42</b>        | <b>57.20</b> | <b>61.1</b>       | 8.99            | <b>8.58</b> | <b>8.79</b> |

\* SimPO still relies on margin-difference style loss, but removed the reference model and added length normalization

# Resolving Likelihood Displacement



NEURAL INFORMATION  
PROCESSING SYSTEMS

informs  
ANNUAL  
MEETING

Table 3: The log-likelihood for preferred and dispreferred responses for 3 independent trials with  $\gamma \in \{0.1, 1\}$  and the default values for all of other parameters. Each cell gives a pair of log-likelihood for preferred and dispreferred responses ( $\log \pi_\theta(\mathbf{y}^+|\mathbf{x}), \log \pi_\theta(\mathbf{y}^-|\mathbf{x})$ ) after one trial of training. The results are indeed different since the perturbations  $\{\mathbf{z}_i\}_{1 \leq i \leq m}$  are different for Trial 1, Trial 2 and Trial 3. However, we find that the log-likelihood for preferred response increase and the log-likelihood for dispreferred response decrease.

| <b>Llama-3-Instruct-8B</b> $(\log \pi_\theta(\mathbf{y}^+ \mathbf{x}), \log \pi_\theta(\mathbf{y}^- \mathbf{x})) = (-46.761, -47.410)$ |                      |                      |                      |
|--|----------------------|----------------------|----------------------|
| $\gamma$   | Trial 1              | Trial 2              | Trial 3              |
| 0.1  | (-46.744, -47.411)   | (-46.760, -47.411)   | (-46.759, -47.410)   |
| 1  | (-46.728, -47.520)   | (-46.743, -47.525)   | (-46.753, -47.517)   |
| <b>Gemma-2-it-9B</b> $(\log \pi_\theta(\mathbf{y}^+ \mathbf{x}), \log \pi_\theta(\mathbf{y}^- \mathbf{x})) = (-133.122, -134.557)$     |                      |                      |                      |
| $\gamma$   | Trial 1              | Trial 2              | Trial 3              |
| 0.1  | (-133.122, -134.557) | (-133.122, -134.557) | (-133.121, -134.557) |
| 1  | (-133.059, -134.562) | (-133.122, -134.564) | (-133.112, -134.565) |

# Memory + Computational Efficiency

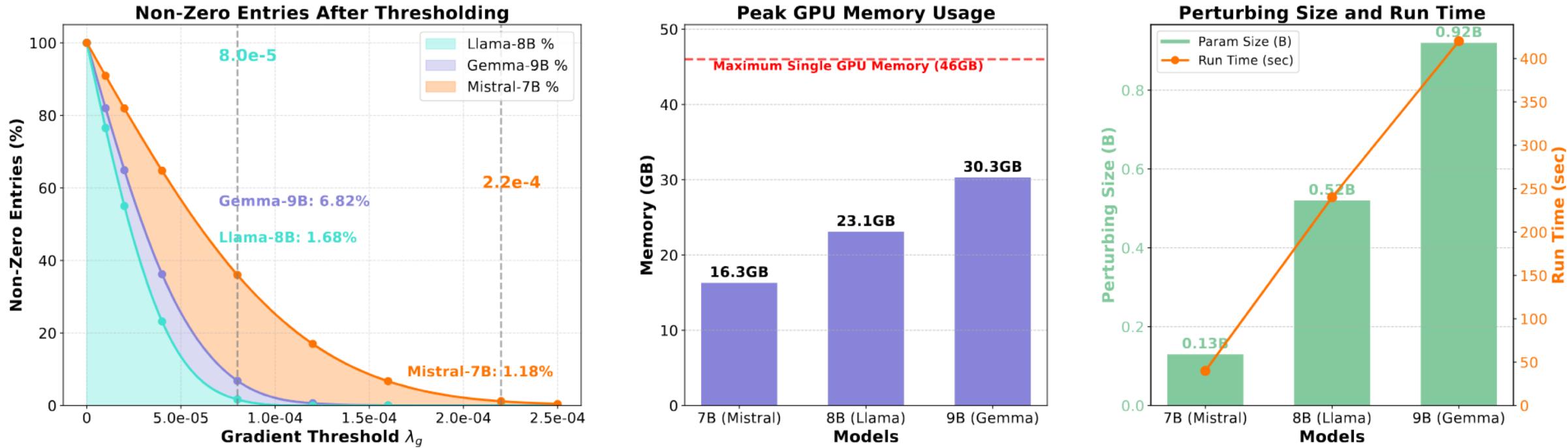


Figure 1: (Left) Percentage of non-zero entries in the final gradient across different gradient entry threshold  $\lambda_g$ ; (Middle) Peak GPU memory usage across three models used in all experiments; (Right) Size of parameter space (output layer) in the comparison oracle perturbations and the run time for completing 600 perturbations using 30 NVIDIA A40 GPUs are shown.

\* Running DPO and SimPO on Llama-8B-Instruct requires **77GB** and **69GB**, respectively, while ComPO only requires **23GB**

# Scalability: hyperparameter

## I. Number of random perturbation in each iteration:

**Table 4:** Effect of the number of perturbations  $m$  on model performance. We report WR and LC on AlpacaEval 2; each entry includes the results of mean performance and standard deviation over 5 consecutive runs; the best run performance is shown in the parentheses.

| Perturbation ( $m$ )     | 800                      | 1600                     | 3300                              | 5400                              |
|--------------------------|--------------------------|--------------------------|-----------------------------------|-----------------------------------|
| <b>AlpacaEval 2-WR %</b> | $17.32 \pm 0.86$ (17.94) | $17.50 \pm 0.65$ (18.32) | $19.21 \pm 0.58$ ( <b>20.25</b> ) | $19.69 \pm 0.36$ (20.07)          |
| <b>AlpacaEval 2-LC %</b> | $24.72 \pm 1.02$ (25.12) | $25.02 \pm 0.91$ (26.17) | $25.91 \pm 0.95$ (27.14)          | $26.49 \pm 0.81$ ( <b>27.20</b> ) |

## 2. Number of layer perturbed in LLM:

**Table 5:** Multi-layer perturbation improves performance. We report WR and LC on AlpacaEval 2 and WR on Arena-Hard; entries are mean  $\pm$  std over 5 runs, with the best run in parentheses.

| Layers perturbed (# params) | AlpacaEval 2-WR %        | AlpacaEval 2-LC %        | Arena-Hard (GPT 4.1)-WR % |
|-----------------------------|--------------------------|--------------------------|---------------------------|
| 1 (0.13B)                   | $17.50 \pm 0.65$ (18.32) | $25.02 \pm 0.91$ (26.17) | $10.80 \pm 0.21$ (11.0)   |
| 3 (0.25B)                   | $18.19 \pm 0.81$ (19.38) | $26.00 \pm 0.89$ (27.09) | $11.26 \pm 0.36$ (11.7)   |

### 3. Number of noisy pairs used in training:

**Table 7:** Results on scaling the number of noisy preference pairs used in the training.

| <b>Number of noisy pairs</b> | <b>AlpacaEval 2-WR %</b> | <b>AlpacaEval 2-LC %</b> | <b>Arena-Hard (GPT 4.1)-WR %</b> |
|------------------------------|--------------------------|--------------------------|----------------------------------|
| 100                          | $19.21 \pm 0.58$ (20.25) | $25.91 \pm 0.95$ (27.14) | $11.02 \pm 0.13$ (11.2)          |
| 300                          | $20.07 \pm 0.99$ (21.35) | $26.28 \pm 0.81$ (27.59) | $11.76 \pm 0.30$ (12.1)          |

### 4. Ablation on gradient threshold $\lambda_g$ :

**Table 6:** Effect of gradient threshold  $\lambda_g$  (transposed). Results are WR and LC on AlpacaEval 2; entries are mean  $\pm$  std over 5 runs, with the best run in parentheses.

| $\lambda_g$                                   | 0                        | $4 \times 10^{-5}$       | $1.8 \times 10^{-4}$     | $2.2 \times 10^{-4}$     | $2.5 \times 10^{-4}$     |
|---|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| <b>Percentage of gradient entries updated</b> | 100%                     | 63%                      | 6%                       | 1%                       | 0.15%                    |
| <b>AlpacaEval 2-WR %</b>                      | $15.72 \pm 0.77$ (16.34) | $16.02 \pm 0.69$ (16.69) | $19.02 \pm 0.62$ (20.15) | $19.21 \pm 0.58$ (20.25) | $16.10 \pm 0.11$ (16.21) |
| <b>AlpacaEval 2-LC %</b>                      | $23.42 \pm 1.03$ (24.28) | $24.01 \pm 0.91$ (25.10) | $26.06 \pm 0.81$ (27.27) | $25.91 \pm 0.95$ (27.14) | $23.82 \pm 0.23$ (24.00) |

# Q&A - Thanks for joining!



Paper: arxiv.org/pdf/2505.05465

Email: [lc3826@columbia.edu](mailto:lc3826@columbia.edu)

Twitter: [@PeterLauLukCh](https://twitter.com/PeterLauLukCh)