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

NEURAL INFORMATION PROCESSING SYSTEMS

informs ANNUAL MEETING

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

NYU | STERN

DAMO ALIBABA DAMO ACADEMY

LLM Alignment from Human Preference Feedback

NEURAL INFORMATION PROCESSING SYSTEMS

informs ANNUAL MEETING

Input prompt:

I am working on **optimization**. Please introduce Adam to me.

Output A:

Who is Adam?

Output B:

Adam is a popular optimizer widely applied to stochastic optimization.

Output C:

Adam is a first-order optimizer that blends momentum with per-parameter adaptive learning rates: it keeps exponential moving averages of the gradients (mean) and their squares (variance), applies bias correction for early steps...

Human Preference Feedback: C ≻ B ≻ A

A – Factual mistake

B – Correct, but too higher-level

C – Correct, detailed and informative

preferred: detailed & concise, honest & harmless, informative & helpful, clarity & structured, respectful & accurate, User-centered

non-preferred: misaligned with prompt, harmful info, verbose & lengthy, superficial & unclear, unsafe & dishonest, User-disobeyed

RLHF: Reinforcement Learning from Human Feedback

Goal: encourage preferred output & discourage non-preferred output

Preference Dataset

$$\mathcal{D}(x, y^+, y^-)$$

Preference pairs:

- x: prompt
- $y+$: preferred response
- $y-$: non-preferred response

Stage 1
Human Annotation

Reward Model

$$r_\phi(x, y)$$

Reward model r_ϕ:

Trained to provide:
- high reward for preferred one
- low reward for non-preferred response

Stage 2
Training Reward Model

Reinforcement Learning

$$\pi^*(y \mid x)$$

Optimal policy π^*:

Trained to:
- Maximize reward $r_\phi(x, y)$
- Policy guided rollout towards human preferred response

Stage 3
PPO Policy Update

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

<u>DPO loss</u>: a reparameterization from RLHF objective

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

Likelihood margin:
We want larger margin between each winning/losing pair

$\pi(y^+|x) = -5, \qquad \pi(y^-|x) = -100$

Large margin $\Rightarrow$ Stronger signal

$\pi(y^+|x) = -99, \qquad \pi(y^-|x) = -100$

Smaller margin $\Rightarrow$ Weaker/noisy signal
DPO cannot utilize it, simply discarded…

Can we extract useful information from these noisy preference pairs?

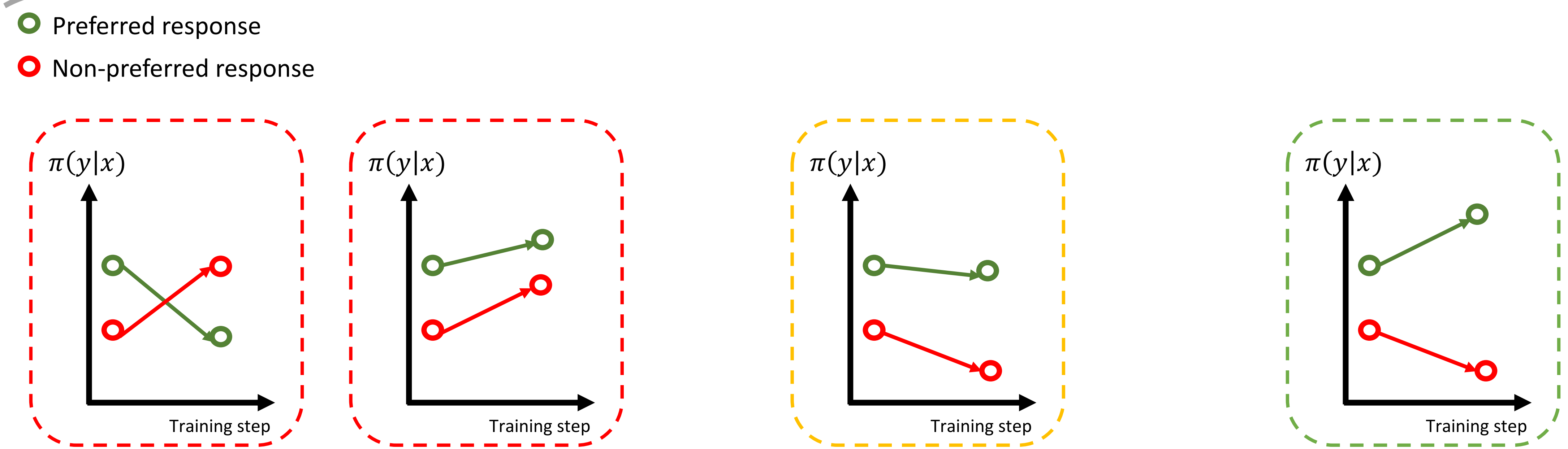
ComPO: Zeroth-order, Comparison-based method

NEURAL INFORMATION PROCESSING SYSTEMS
informs ANNUAL MEETING

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

$\Longrightarrow$ failed when such difference is numerically small, especially under noisy pairs

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



(a) Misalignment

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

Expected

Comparison Oracles

Given that $f : \mathbb{R}^d \to \mathbb{R}$ 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* $\mathcal{C}_f(\theta, \theta') : \mathbb{R}^d \times \mathbb{R}^d \to \{+1, -1\}$ *a comparison oracle for function* f *if*

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

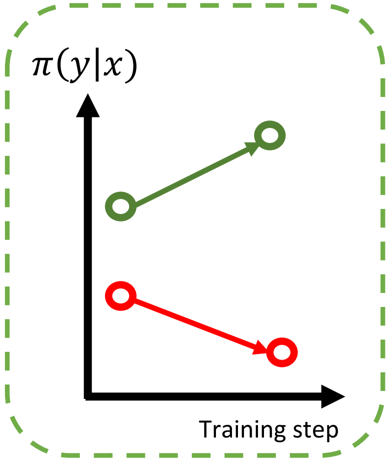
Goal: minimizing f, but only get access to implicit comparison between two sets of model parameter θ

tl;dr "**-1 returned**: ***θ'*** **is better than** ***θ*** **for** ***f***"

Preference Comparison Oracles

Definition 3.1 *We say* $\mathcal{C}_\pi(\theta, \theta') : \mathbb{R}^d \times \mathbb{R}^d \to \{+1, -1\}$ *a preference comparison oracle for the model* π_θ *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):

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

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

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

Choose 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

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

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

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

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

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

Step 2: Derive simplified gradient
Retrieve with closed form

Step 3: Apply gradient-entry threshold λ_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 $\theta_1 = [\bar{\theta}_1; \theta_t^{\mathrm{o}}] \in \mathbb{R}^d$, 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** $t = 1, 2, \ldots, T$ **do**

3: Draw m i.i.d. samples uniformly from a unit sphere in $\mathbb{R}^{d^{\mathrm{o}}}$, i.e., $\{\mathbf{z}_i\}_{1 \leq i \leq m}$.

4: Compute $y_i = \mathcal{C}_\pi([\bar{\theta}_1; \theta_t^{\mathrm{o}}], [\bar{\theta}_1; \theta_t^{\mathrm{o}} + r\mathbf{z}_i])$ for $i = 1, 2, \ldots, m$.

5: Compute $\hat{\mathbf{g}}_t^{\mathrm{o}} = \frac{\sum_{i=1}^m y_i \mathbf{z}_i}{\|\sum_{i=1}^m y_i \mathbf{z}_i\|}$ and clip $\hat{\mathbf{g}}_t^{\mathrm{o}}$ by zeroing out the entries whose magnitude is less than $\boxed{\lambda_g}$.

6: Compute $\theta_{t+1}^{\mathrm{o}} = \theta_t^{\mathrm{o}} - \frac{\gamma|\{i : y_i = -1\}|}{m} \hat{\mathbf{g}}_t^{\mathrm{o}}$ if $\frac{|\{i : y_i = -1\}|}{m} > \boxed{\lambda}$ and $\theta_{t+1}^{\mathrm{o}} = \theta_t^{\mathrm{o}}$ otherwise.

Gradient entry threshold

Stepsize threshold (oracle feedback)

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

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

* $\delta = 3$ 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 $\text{DPO}_{\text{clean}}$+ComPO and present the best trail 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	**5.38**	**5.79**	24.14	16.71	**14.4**	6.28	5.42	5.86
$\text{DPO}_{\text{clean}}$	9.41	6.52	3.0	6.18	5.22	5.70	23.89	16.15	14.2	6.11	5.34	5.73
$\text{DPO}_{\text{clean}}$+ComPO	**11.66**	**6.55**	**3.2**	**6.22**	5.32	5.77	**26.17**	**18.32**	10.5	**7.78**	**7.63**	**7.69**

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	**12.1**	6.61	5.85	6.23	32.59	31.99	22.9	8.30	7.55	7.93
$\text{DPO}_{\text{clean}}$	4.28	9.81	12.0	**6.64**	6.01	6.33	32.92	32.42	22.9	8.26	7.63	7.94
$\text{DPO}_{\text{clean}}$+ComPO	**5.39**	**10.93**	**12.1**	6.60	**6.28**	**6.44**	**35.79**	**35.03**	**23.1**	**8.39**	**7.71**	**8.05**

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

$\text{DPO}_{\text{clean}}$: Running DPO while excluding those noisy pairs

Augmenting DPO variants – SimPO (Meng et al., 2024)

NEURAL INFORMATION PROCESSING SYSTEMS

informs ANNUAL MEETING

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

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

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

Resolving Likelihood Displacement

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 trail 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.

Llama-3-Instruct-8B $(\log \pi_\theta(\mathbf{y}^+\|\mathbf{x}), \log \pi_\theta(\mathbf{y}^-\|\mathbf{x})) = (-46.761, -47.410)$			
γ	Trial 1	Trial 2	Trial 3
0.1	$(-46.744,\ -47.411)$	$(-46.760,\ -47.411)$	$(-46.759,\ -47.410)$
1	$(-46.728,\ -47.520)$	$(-46.743,\ -47.525)$	$(-46.753,\ -47.517)$
Gemma-2-it-9B $(\log \pi_\theta(\mathbf{y}^+\|\mathbf{x}), \log \pi_\theta(\mathbf{y}^-\|\mathbf{x})) = (-133.122, -134.557)$			
γ	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

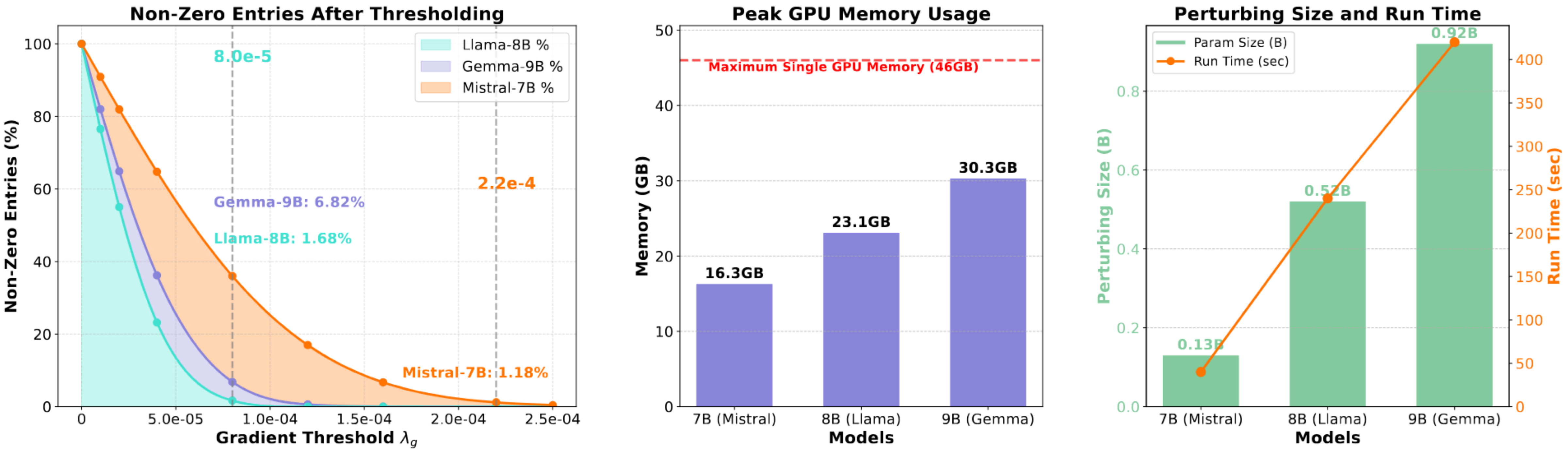


Figure 1: (Left) Percentage of non-zero entries in the final gradient across different gradient entry threshold λ_g; (Middle) Peak GPU memory usage across three models used in all experiments; (Right) Size of parameter space (output layer) in the comparison oracle perturbations and the run time for completing 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

1. 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**
AlpacaEval 2-WR %	17.32 ± 0.86 (17.94)	17.50 ± 0.65 (18.32)	19.21 ± 0.58 (**20.25**)	19.69 ± 0.36 (20.07)
AlpacaEval 2-LC %	24.72 ± 1.02 (25.12)	25.02 ± 0.91 (26.17)	25.91 ± 0.95 (27.14)	26.49 ± 0.81 (**27.20**)

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 ± 0.65 (18.32)	25.02 ± 0.91 (26.17)	10.80 ± 0.21 (11.0)
3 (0.25B)	18.19 ± 0.81 (19.38)	26.00 ± 0.89 (27.09)	11.26 ± 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.

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

4. Ablation on gradient threshold λ_g:

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

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

Q&A – Thanks for joining!

Paper: arxiv.org/pdf/2505.05465

Email: lc3826@columbia.edu

Twitter: @PeterLauLukCh