Test-time Preference Optimization: On-the-fly Alignment via Textual Feedback

Yafu Li, Xuyang Hu, Xiaoye Qu, Linjie Li, Yu Cheng†

Shanghai AI Laboratory, University of Washington, The Chinese University of Hong Kong

Paper Link

ICML International Conference On Machine Learning

上海人工智能实验室 Shanghai Artificial Intelligence Laboratory

Motivation

Current preference optimization (RLHF, DPO) occurs during training.

- ➢ Requires costly retraining for new domains, regulations, or preferences.
- ➢ Once deployed, models are static and cannot adapt to evolving user needs.

Goal:

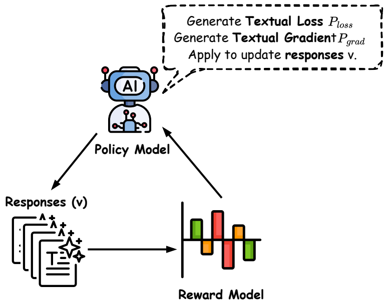
Enable preference alignment at inference time, with minimal compute and no parameter updates.

$$\log p_{\boldsymbol{\theta}}(y|x;\boldsymbol{\varphi})$$

Parameters Context

- ➢ DPO/RLHF: update $\boldsymbol{\theta}$
- ➢ TPO: update $\boldsymbol{\varphi}$

Test-time Reinforcement Learning via Textual Feedback



Initialization: policy model $\mathcal{M}$, reward model $\mathcal{R}$, user query x

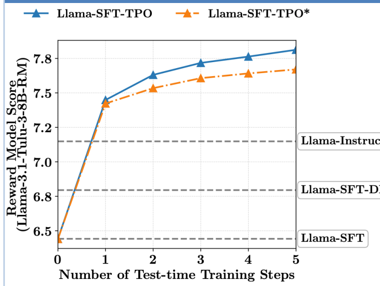
- Sample **N** candidate responses $v_1, v_2, \dots, v_N \leftarrow \mathcal{M}(x)$
- Score with reward model $\mathcal{R}$; store $(v_i, \mathcal{R}(v_i))$ in cache $\mathbb{C}$

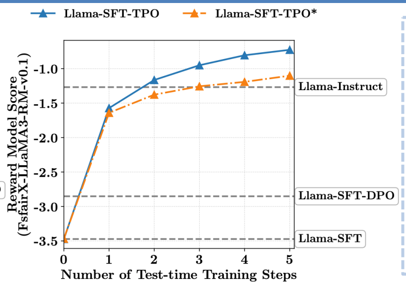
Iterate for t=1…D

- **Select** the best and worst responses from $\mathbb{C}$
- $\mathcal{M}$: **Generate** *textual loss* comparing "best" and "worst"
- $\mathcal{M}$: **Generate** *textual gradient* ($\boldsymbol{\varphi}$) suggesting how to improve "best" further.
- $\mathcal{M}$: **Update** responses; **score** with $\mathcal{R}$ and add to cache $\mathbb{C}$

Output Return highest-scoring response in $\mathbb{C}$

Aligning Preferences during Inference





TPO progressively improves alignment over test-time steps:

- both unaligned and aligned models
- across different reward models

Benchmark Performance

Model	AlpacaEval 2 LC(%)	AlpacaEval 2 WR(%)	Arena-Hard	HH-RLHF	BeaverTails	XSTest	MATH-500
Llama-3.1-70B-DPO	32.3	23.1	50.4	-2.8	-6.7	89.8	63.4
Llama-3.1-70B-Instruct	36.9	34.9	59.0	-0.5	-6.4	88.7	66.4
Llama-3.1-70B-SFT	27.8	16.8	44.1	-4.1	-7.2	87.8	61.8
w/ TPO (D2-N5) †	33.2	39.5	70.5	0.1	**-4.1**	89.8	70.0
w/ TPO (D2-N5) ⋆	33.0	40.5	69.7	-0.6	-4.8	**90.4**	71.2
w/ TPO (D5-N20) ⋆	**37.8**	**55.7**	**77.5**	**0.4**	**-4.1**	89.6	**71.8**

TPO on the unaligned model (after SFT **without** training-time alignment).

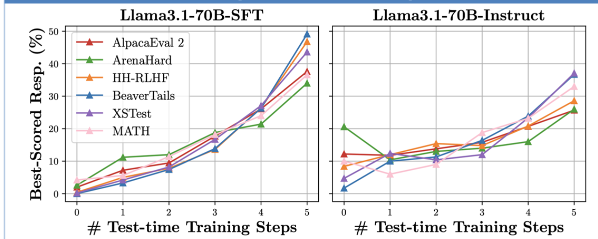
Model	AlpacaEval 2 LC(%)	AlpacaEval 2 WR(%)	Arena-Hard	HH-RLHF	BeaverTails	XSTest	MATH-500
Llama-3.1-70B-Instruct	36.9	34.9	59.0	-0.5	-6.4	88.7	66.4
w/ TPO (D2-N5)	39.1	48.5	69.5	**1.3**	-3.6	89.6	**71.6**
Mistral-Small-Instruct-2409	45.7	38.5	53.8	-0.4	-5.2	87.1	57.6
w/ TPO (D2-N5)	**53.4**	**60.5**	**72.2**	1.1	**-3.4**	**90.7**	62.2

TPO on the aligned models (after training-time alignment).

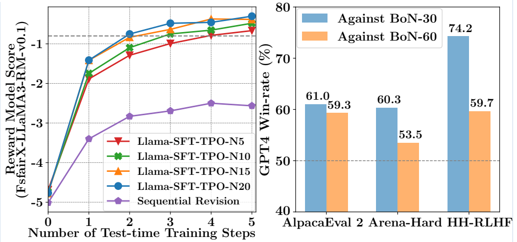
On the unaligned model, TPO (D5-N20) outperforms DPO and Instruct (e.g., 77.5% WR on Arena-Hard, 71.8 on MATH-500).

On aligned models, TPO further boosts performance with minimal extra compute.

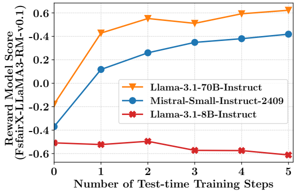
Test-time Scaling: Combining Parallel Sampling with Sequential Revision



Unaligned models benefit from more iterative refinement as better responses emerge from later TPO steps.



TPO-D2-N5 beats BoN-30/60 with less samples showing the efficiency of iterative revision.



TPO requires instruction-following ability, as models must accurately interpret and act on textual feedback to align effectively.

Scaling computing from training-time to test-time

- ➢ LLAMA-3.1-70B-**DPO**: **72,840 PFLOPs**
- ➢ LLAMA-3.1-70B-**TPO**: **9.3 PFLOPs (0.013%)**

Contact & Code

yafuly

yafuly@gmail

yafuly/TPO