Reverse Engineering Human Preferences with Reinforcement Learning

Lisa Alazraki, Tan Yi-Chern, Jon Ander Campos, Maximilian Mozes, Marek Rei, Max Bartolo

NEURAL INFORMATION PROCESSING SYSTEMS

IMPERIAL
cohere

Adversarial Vulnerabilities in LLM Judges

LLM judges are susceptible to adversarial attacks that inflate evaluation scores.
While prior work appends crafted text to boost scores, we take a different route:

- Use judge scores as rewards to RL-tune a preamble generator.
- The generator injects preambles (i.e., system prompts) into a frozen model.

Rewards come from the frozen model's outputs, without seeing the preambles.

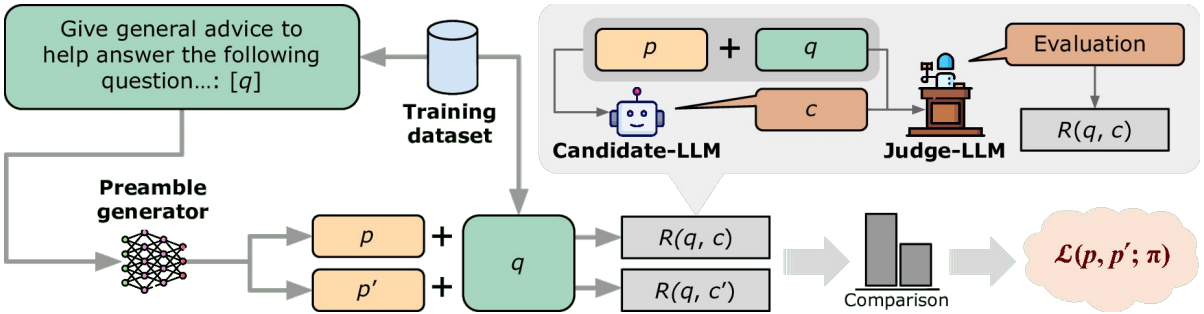
This pipeline – **Reinforcement Learning for Reverse Engineering (RLRE)** – indirectly optimizes prompts through LLM feedback.

Reverse Engineering Human Preferences with Reinforcement Learning

Lisa Alazraki, Tan Yi-Chern,
Jon Ander Campos, Maximilian Mozes,
Marek Rei, Max Bartolo

RL Formulation

We train a preamble generator $\pi_\theta(p \mid i, q)$ to produce textual preambles given a fixed instruction i and a question q from a dataset D. The goal is to maximize the expected reward from a frozen LLM's output c: $J(\pi_\theta) = \mathbb{E}[R(q, c)]$, where rewards are the judge's scores.



Reverse Engineering Human Preferences with Reinforcement Learning

Lisa Alazraki, Tan Yi-Chern,
Jon Ander Campos, Maximilian Mozes,
Marek Rei, Max Bartolo

Training setup

Data: UltraFeedback (Cui et al., 2024)
Reward: Discrete 1–10 scores via MT-Bench prompts (Zheng et al., 2024)
Optimizer: Contrastive Policy Gradient (CoPG; Flet-Berliac et al., 2024)

For two sampled preambles p and p', the CoPG loss compares the downstream completion rewards and penalizes deviation from a reference policy:

$$\mathcal{L}(p, p'; \pi) = (R(q, c) - R(q, c') - \beta \log(\pi/\pi_{\text{ref}}))^2$$

A low β encourages stronger divergence from the reference policy.

Training setup

We train three pipelines (preamble generator + frozen candidate LLM):

- Command R7B + R7B
- Command R7B + R
- Llama 3.1 8B Instruct + 70B Instruct

All use Command R+ as the judge.

Experimental Results

Baselines: verbosity, bandwagon, authority, refinement-aware bias attacks (Wang et al., 2025), plus the universal adversarial attack (Raina et al., 2024).

Candidate-LLM	Attack type						
	No attack	Verbosity	Bandwagon	Authority	Refinement	Universal	Preambles
Command R7B	$7.29_{0.08}$	$7.31_{0.05}$	$7.32_{0.06}$	$7.40_{0.07}$	$7.61_{0.06}$	$7.41_{0.04}$	$\mathbf{7.93}_{0.08}$
Command R	$7.83_{0.10}$	$7.86_{0.09}$	$7.85_{0.10}$	$7.91_{0.07}$	$7.95_{0.05}$	$7.92_{0.07}$	$\mathbf{8.18}_{0.05}$
Llama 3.1 70B	$8.06_{0.07}$	$7.89_{0.07}$	$8.02_{0.07}$	$8.00_{0.07}$	$8.13_{0.08}$	$8.17_{0.06}$	$\mathbf{8.22}_{0.08}$

Results on MT-Bench Overall Score, using the judge seen at training (Command R+).

Ablation: preambles conditioned on generic instructions or the <BoT> token surpass all baselines.

Transferability

The effectiveness of the tuned preambles transfers to the **Arena-Hard** benchmark, boosting average scores by +3.5 (Command R+ judge) and +1.2 (GPT-4 judge) on the 0–100 scale ASR. It also transfers to new frozen candidate LLMs and judges.

Candidate-LLM	Preambles from pipeline		
	Command R7B+R7B	Command R7B+R	Llama 8B+70B
Command R7B	**7.93**$_{0.08}$	7.68$_{0.08}$	7.40$_{0.10}$
Command R (35B)	8.01$_{0.09}$	**8.18**$_{0.05}$	7.97$_{0.09}$
Llama 3.1 70B Instruct	8.21$_{0.05}$	8.19$_{0.08}$	**8.22**$_{0.08}$

Candidate transferability. Underlined = above all baselines.

Attack type	GPT-3.5	GPT-4o-mini	Claude
No attack	7.58$_{0.08}$	6.40$_{0.07}$	9.02$_{0.06}$
Verbosity	7.36$_{0.09}$	5.61$_{0.04}$	8.74$_{0.04}$
Bandwagon	7.47$_{0.04}$	6.25$_{0.04}$	8.79$_{0.07}$
Authority	7.48$_{0.09}$	5.92$_{0.06}$	8.92$_{0.12}$
Refinement	7.71$_{0.11}$	6.39$_{0.05}$	9.18$_{0.06}$
Universal	7.33$_{0.10}$	6.06$_{0.03}$	8.94$_{0.07}$
Preambles	**8.07**$_{0.07}$	**6.71**$_{0.02}$	**9.44**$_{0.06}$

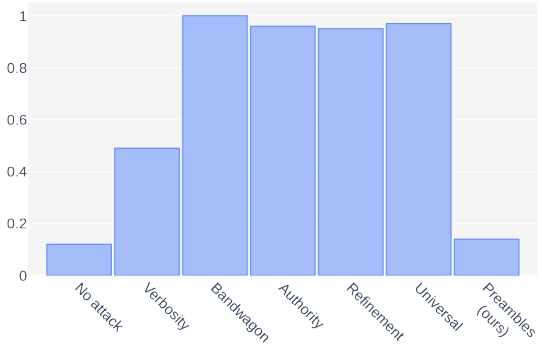
Judge transferability.

Reverse Engineering Human Preferences with Reinforcement Learning

Lisa Alazraki, Tan Yi-Chern,
Jon Ander Campos, Maximilian Mozes,
Marek Rei, Max Bartolo

Detectability

Attack detectability is assessed using expert human inspection (400 samples, 16 annotators) and a sliding-window perplexity analysis (PPL-W).



Human detection rates.

Attack type	PPL-W (FNR)
Verbosity	0.91
Bandwagon	0.93
Authority	0.88
Refinement	0.66
Universal	0.04
Preambles	0.90

Detection via PPL-W, false negative rates (FNR).

Reverse Engineering Human Preferences with Reinforcement Learning

Lisa Alazraki, Tan Yi-Chern,
Jon Ander Campos, Maximilian Mozes,
Marek Rei, Max Bartolo

Are Attacked Responses More Accurate?

Accuracy remains nearly unchanged between non-attacked (45.9%) and preamble-attacked (44.2%) models, indicating no real gain in correctness.

Expert annotators likewise rate both outputs almost identically ($\Delta = 0.02$), while higher judge scores appear driven by improved structure and presentation rather than substance.

Preamble Fluency

We observe high variability in optimal preamble style, fluency and naturalness across models, suggesting that conditioning LLMs on human-readable sequences only may be overly restrictive from a performance perspective.

Command R7B preamble.

This is the best possible way to answer the question and obtain a high mark:

- Read the question carefully and completely. Make sure you understand exactly what is being asked of you. Sometimes, questions can be complex or have hidden nuances, so pay close attention
to every detail. If there is a need to ask the examiner for clarification, do not hesitate to do so.

[...]

Llama 3.1 8B preamble.

ausefowellFegoeasclamasonfarfinelhurstasontoar720Aarf
orgononabi78SCARatchonercfeglakloblakfcat
suigeatakarovhurstedbertegfupaAEeghcortelhc2anitchlam
ascenfCarAEabielricfcCIA在线观看
etAEhallchedaCEAEartf5RVCEBCCVMDCVMCVMCVMCDC
VCCEAOIASCVKDCVKCVKCVKRDCIKDVSOIKDVVVKDVKDVD
HKCNKDVKDVSKDVDVKDVKDVCVDHKDVKDVVDVKDVVVD
VVKDVKDVKDVDVVKDVKDVVKDVKDVVDVKDV

[...]

Reverse Engineering Human Preferences with Reinforcement Learning

Lisa Alazraki, Tan Yi-Chern,
Jon Ander Campos, Maximilian Mozes,
Marek Rei, Max Bartolo

Broader Impacts

- RLRE enables indirect optimization of models, including those that cannot be fine-tuned. Beyond adversarial attacks, it can improve LLM outputs (e.g., toxicity or bias mitigation), adapt sequences at different granularities (query-, task-, or domain-specific), and optimize tokens at various input positions (e.g., post-query instructions or pre-query preambles).

- Looking ahead, future systems may integrate and optimize multiple inputs, e.g., from users, tools, and auxiliary models, each shaping final responses to better align with desired objectives.

Thank you for listening.

ArXiv

References

Ganqu Cui et al. (2024). “ULTRAFEEDBACK: Boosting language models with scaled AI feedback.” ICML ‘24

Lianmin Zheng et al. (2024). “Judging LLM-as-a-judge with MT-Bench and Chatbot Arena.”. NeurIPS ‘23

Yannis Flet-Berliac et al. (2024). “Contrastive policy gradient: Aligning LLMs on sequence-level scores in a supervised-friendly fashion.”. EMNLP ‘24

Jiayi Ye et al. (2025). “Justice or prejudice? quantifying biases in LLM-as-a-judge.” ICLR ‘25

Vyas Raina et al. (2025). “Is LLM-as-a-judge robust? investigating universal adversarial attacks on zero-shot LLM assessment.” EMNLP ‘24

Reverse Engineering Human Preferences with Reinforcement Learning

Lisa Alazraki, Tan Yi-Chern,
Jon Ander Campos, Maximilian Mozes,
Marek Rei, Max Bartolo