URPO: A Unified Reward & Policy Optimization Framework for Large Language Models

Songshuo Lu, Hua Wang, Zhi Chen, Yaohua Tang

Moore Threads AI, China
tangyaohua28@gmail.com

Abstract

Large-scale alignment pipelines typically pair a policy model with a separately trained reward model whose parameters remain frozen during reinforcement learning (RL). This separation creates a complex, resource-intensive pipeline and leads to a performance ceiling. We propose a novel framework, Unified Reward & Policy Optimization (URPO), that unifies instruction-following ("player") and reward modeling ("referee") into a single model and a single training phase. Our method recasts all alignment data-including preference pairs, verifiable reasoning, and open-ended instructions-into a unified generative format optimized by a single Group-Relative Policy Optimization (GRPO) loop. This enables the model to learn from ground-truth preferences and verifiable logic while simultaneously generating its own rewards for open-ended tasks. Experiments on the Qwen2.5-7B model demonstrate that URPO significantly outperforms a strong baseline using a separate generative reward model, boosting the instruction-following score on AlpacaEval to 44.84 and achieving a 36% relative improvement on the challenging AIME reasoning benchmark. Furthermore, URPO cultivates a superior internal evaluator as a byproduct of training, achieving a RewardBench score of 85.15 and surpassing the dedicated reward model it replaces (83.55). By eliminating the need for a separate reward model and fostering a co-evolutionary dynamic, URPO presents a simpler, more efficient, and more effective path towards robustly aligned language models.

Code — https://github.com/MooreThreads/URPO

Introduction

The advent of Large Language Models (LLMs) such as GPT-4 (Achiam et al. 2023) and Llama (Touvron et al. 2023) has revolutionized the field of artificial intelligence, demonstrating remarkable capabilities in understanding and generating human-like text. Post-training alignment, a process that refines the model's outputs to better align with specific instructions and human preferences, emerges to be a key technique for further enhancing the capabilities of LLMs. During this process, RLHF (Christiano et al. 2017; Ziegler et al. 2019) is adopted as a primary approach to align the outputs of LLMs (Ouyang et al. 2022). The conventional RLHF pipeline is typically composed of two decoupled steps, where a reward model (the "referee") is first trained in a supervised manner with direct human feedback, and then this model is taken as a static reward function to evaluate a policy model (the "player") through an optimization algorithm, such as proximal policy optimization (PPO) (Schulman et al. 2017).

This separation of roles leads to three fundamental limitations. First, the pipeline is **cumbersome and inefficient**, requiring the maintenance of multiple models and complex training stages. The problem is exacerbated by the recent addition of GRPO (Shao et al. 2024) for reasoning tasks (Guo et al. 2025). Second, the cooperation between the dynamic policy model and the static reward model creates a **"competence mismatch"**, i.e. an ever-stronger player is judged by a fixed referee, which yields a performance ceiling. Third, the isolated training process often produces **data silos**, preventing potential synergies between the instruction and preference datasets.

To address these challenges, we ask a foundational question: *Could a single model learn to act as both a player and a referee role simultaneously?* Our paper is a quest driven by this simple question. It proposes a ground-up redesign of the existing RHLF pipeline via a unified model paradigm that allows the generative and evaluative capabilities to co-evolve within a single training loop, therefore creating a virtuous cycle of self-improvement.

To fulfill the redesign, we propose Unified Reward & Policy Optimization (**URPO**), a novel training methodology that unifies three distinct data types, verifiable reasoning problems (e.g. math, code), open-ended instructions, and preference data, into a single training batch that fits the GRPO training, as illustrated in Figure 1. We recast preference triples in reward modeling into an N-way ranking prompt, reward the model with Kendall's τ correlation against ground-truth orderings, and interleave rule-verifiable reasoning tasks to ground the policy in factual correctness. For each open-ended instruction, the model samples k rollouts in GRPO training, combines them into a single reward question and prompts itself to rank them, and feeds the induced group rewards to GRPO training. This effectively helps us transform the entire post-training into a single cohesive GRPO training process.

Our contributions are threefold:

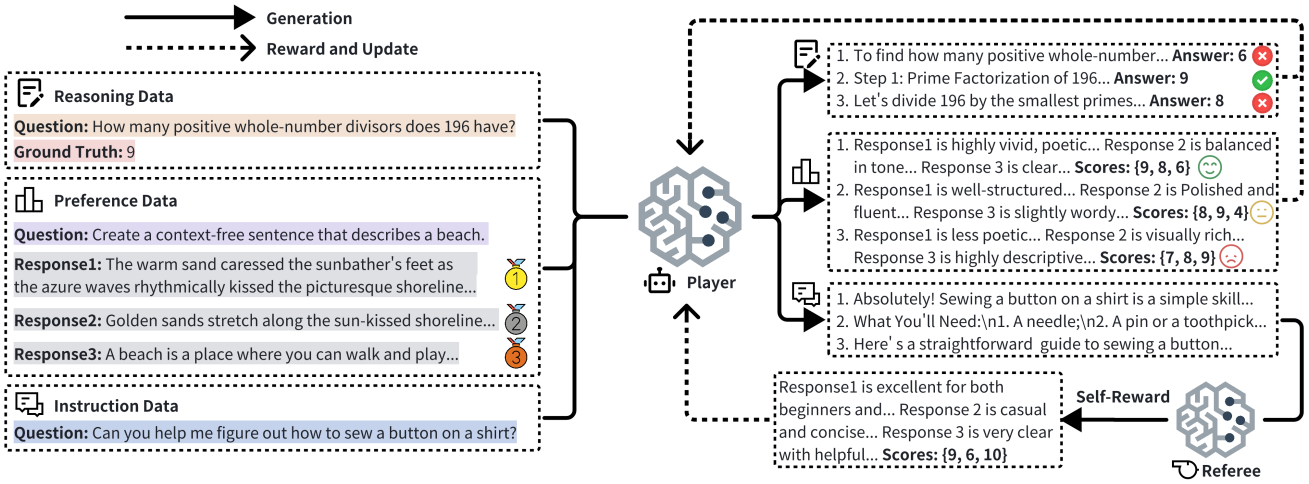

Figure 1: URPO training loop. A single large language model simultaneously plays the roles of *player* (Policy Model) and *referee* (Reward Model). Each mixed training batch contains three data types: (a) rule-verifiable reasoning problems with ground-truth answers, (b) preference triples with N-way ranking, and (c) open-ended instructions. The model first generates candidate responses, then evaluates them with either ground-truth checks or self-assessed ranking rewards before refining its parameters through GRPO, allowing its generation and evaluation capabilities to co-evolve.

- **A Simpler and More Effective Unified Alignment Framework.** We propose URPO, an elegant single-model, single-process framework that redesigns and simplifies the standard reinforcement learning alignment pipeline. Our experiments on Qwen2.5-7B show that URPO significantly outperforms the baseline guided by a separately trained generative reward model. It demonstrates superior performance across the board, boosting the AlpacaEval score to **44.84** and achieving a **36% relative improvement** on the difficult AIME reasoning benchmark (16.46 vs. 12.08).
- **A Smarter, Self-Taught Referee.** With URPO, the model learns to judge its own answers while it learns to generate them simultaneously. This process creates an internal referee that grows smarter over time, overcoming the limitations of a static, pre-trained reward model. Our model becomes a better judge than a specialized reward model that is trained separately for the same task, achieving a higher score on the RewardBench benchmark (**85.15** vs. 83.55).
- **Holistic Improvement Through Data Synergy.** We propose a unified data management approach and perform extensive experiments to evaluate its effectiveness through reasonable data mixing. Our ablation studies show that a balanced data mixture is essential for building a well-rounded model.

Related Work

Reinforcement Learning from Human (and AI) Feedback

Early work established the promise of preference-based fine-tuning by training a reward model on human rankings and optimizing a policy with RL, showing strong gains on tasks such as summarization (Ziegler et al. 2019; Stiennon et al. 2020). This recipe was systematized in InstructGPT (Ouyang et al. 2022), which follows a now-standard multi-stage pipeline: supervised fine-tuning (SFT), reward modeling on pairwise preferences, and policy optimization with PPO under constraints (e.g., KL penalties). While effective, this approach is resource-intensive and sensitive to reward misspecification and optimization instability.

To reduce engineering complexity, subsequent methods recast preference optimization without explicit RL. Direct Preference Optimization (DPO) re-parameterizes the objective so that the optimal policy is obtained via a simple classification-style loss on preference data, avoiding separate reward-model training and sampling-heavy RL updates (Rafailov et al. 2023). Rejection Sampling Fine-Tuning (RFT) similarly bypasses RL by fine-tuning on higher-ranked responses selected by a reward model or heuristic, thereby injecting preference signals through filtered SFT. Another orthogonal thrust lowers labeling costs by replacing or supplementing human raters with AI feedback. Constitutional AI uses model critiques guided by a set of principles to improve harmlessness without direct human preference labels (Bai et al. 2022). Reinforcement Learning from AI Feedback (RLAIF) trains on preferences produced by an auxiliary LLM in place of humans, and a direct-RLAIF variant queries a teacher LLM for rewards on-the-fly, removing the explicit reward-model stage while maintaining competitive alignment performance (Lee et al. 2023). Despite these simplifications, most pipelines still separate the policy from the source of rewards (frozen reward model or external judge).

Self-Critique and Self-Alignment Paradigms

In parallel, self-training paradigms let a model serve as both content generator and evaluator. Iterative refinement approaches prompt the model to critique and revise its own drafts (e.g., Self-Refine (Madaan et al. 2023)) or to reflect over trajectories to improve future behavior (e.g., Reflexion (Shinn et al. 2023)). For factuality, self-alignment methods elicit an LLM's own knowledge to assess the correctness of its outputs and then fine-tune with preference-style objectives (e.g., DPO), substantially reducing hallucinations without human annotations (Zhang et al. 2024).

More formal frameworks unify reward and policy within a single model. Self-rewarding schemes use the model itself as the judge to score or rank its candidate responses and then update the policy to favor higher-judged outputs, often in iterative rounds that also improve the model's judging ability (Yuan et al. 2025). A recognized limitation is that judgment quality can lag behind generation, leading to plateaued gains; meta-rewarding extensions introduce a meta-judge that critiques the judge's own decisions to sharpen evaluative skill (Wu et al. 2024). Complementary work explores intrinsic signals such as model probability or confidence as rewards, showing that self-estimated uncertainty can guide reinforcement-style updates without external labels (Zhao et al. 2025; Zhou et al. 2025).

The above lines of work simplify or automate the RLHF stack but largely preserve a conceptual gap between "player" and "referee." Our URPO instead targets a fully unified perspective while explicitly grounding reward modeling with external human-annotated preference data. This external anchor is intended to prevent degenerate solutions such as self-induced reward hacking and to promote improvements in reward modeling quality alongside instruction following, rather than relying solely on self-generated supervision.

Methodology

Preliminaries

GRPO is a powerful and widely adopted post-training alignment algorithm. The core idea is to sample a group of G responses $\{o_i\}_{i=1}^{G}$ from the policy π_θ and calculate a value-free, group-relative advantage $\hat{A}_i$ for each response based on the reward R_i compared to the group average. The policy is then updated using the canonical GRPO objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\ \{o_i\}_{i=1}^{G}\sim\pi_\theta(\cdot|q)} \left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \Bigg(\min\Big(r_{i,t}(\theta)\hat{A}_i,\ \text{clip}\big(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\big)\hat{A}_i \Big) - \beta\, D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}}) \Bigg) \right] \tag{1}$$

This objective maximizes the advantage $\hat{A}_i$ weighted by an importance ratio $r_{i,t}(\theta)$. The standard formulation typically includes a per-response length normalization ($1/|o_i|$), a clipping mechanism (ε) to stabilize the update, and a KL divergence penalty (β) to regularize the policy against a reference model π_{ref}.

Crucially, in this standard paradigm, the system is fundamentally decoupled. The "player" π_θ and the "referee" r_ϕ that provides the rewards are trained and operate as independent entities. The "referee" for reasoning is hard-coded rules, while the "referee" for instruction-following is a separate, static reward model. This separation leads directly to the shortcomings we identified: a fragmented pipeline, a static competence ceiling for the reward signal, and a lack of internal consistency. These pronounced limitations motivate a fundamental rethinking of the alignment process, leading us to our proposed solution.

URPO

The core innovation of URPO lies in its ability to reformat all training instances into a structure amenable to direct optimization by GRPO, thereby enabling a single model to simultaneously learn generative and evaluative competencies. This approach integrates instruction-following and self-rewarding capabilities within a single LLM, allowing it to learn from verifiable reasoning tasks, open-ended generation, and preference data concurrently. Thus, it eases the traditional GRPO training pipeline and enhances training efficiency.

Formulating Reward Modeling as a Generative GRPO Task The paradigm for training reward models has evolved beyond simple scoring mechanisms. Traditionally, reward models have been trained as separate classifiers or regressors on human preference data (e.g. $\langle q, a_{\text{chosen}}, a_{\text{reject}}\rangle$), learning to output a scalar score, often optimized with a ranking loss like the Bradley-Terry model or BPR loss (Rendle et al. 2012). While effective, this approach creates a static referee, decoupled from the policy model it aims to guide. More recently, the landscape has diversified with the advent of LLM-as-a-Judge (Zheng et al. 2023). This approach leverages a powerful, external large language model (often a proprietary model like GPT-4) to evaluate and score responses. While demonstrating strong correlation with human judgments, this method introduces significant dependencies, either incurring high operational costs through API calls or requiring the fine-tuning of a powerful open-source model on vast amounts of costly, detailed critique data to act as a stand-in judge. Furthermore, emerging research explores training reward models themselves through reinforcement learning paradigms (Liu et al. 2025). However, the optimal methods for integrating these dynamically trained referees back into a stable downstream policy-training pipeline are still an active area of investigation.

URPO fundamentally transforms this landscape by sidestepping the need for any separate or external reward model. It recasts reward modeling as an intrinsic generative task performed by the primary policy model itself. This eliminates the complexities of managing a second model, the costs associated with external judges, and the uncertainties of novel reward training schemes, creating a fully self-contained and co-evolutionary system.

Our framework constructs input prompts by concatenat-

ing the preference data into a multi-response evaluation template, instructing the model to act as an evaluator:

> You are a skilled expert at scoring responses. You will be given a question with multiple responses. Evaluate and score each response.
> `[The Begin of Question]`$\{q\}$`[The End of Question]`
> `[The Begin of Response 1]`$\{y_w\}$`[The End of Response 1]`
> ...
> `[The Begin of Response N]`$\{y_l\}$`[The End of Response N]`
> #### Output Format Requirements ####
> Scores: <the overall comprehensive score of all responses in order, separate by comma in the boxed, e.g., \boxed{x, x} if there exists 2 responses>.

In real model training, the number of reference samples is not restricted to two but can be actually many, provided that a ground-truth-based rank ordering exists. We integrate these preference data into the training batch with the aforementioned prompt through the following GRPO steps:

Rollout: For a given question q and a set of N response candidates $\{a_i\}_{i=1}^N$ (derived from preference data), these are first assembled into a single prompt using the above template. The policy π_θ then generates a group of G evaluation outputs by rolling out this prompt, with each output containing predicted scores for the N responses.

Reward: Each generated evaluation output provides a ranking of the N responses based on the assigned scores. Let $\mathbf{s} = (s_1, \ldots, s_N)$ be the predicted scores for responses $\{a_i\}_{i=1}^N$, yielding a predicted ranking vector $\mathbf{r}$. We compare this predicted ranking with the ground-truth ranking $\mathbf{g}$ (derived from human/AI preferences) using Kendall's τ correlation coefficient as the scalar reward signal:

$$R(q, \{a_i\}_{i=1}^N) = \tau_K(\mathbf{r}, \mathbf{g}) = \frac{2}{N(N-1)} \sum_{1 \le i < j}^{N} \text{sign}\big[(s_i - s_j)(g_i - g_j)\big] \tag{2}$$

where $\text{sign}(x) \in \{-1, 0, 1\}$ is the sign function. This reward $R(q) \in [-1, 1]$ is 1 for perfect agreement and -1 for perfect disagreement.

GRPO Update: These Kendall's τ rewards are then used within the GRPO objective to update the policy, encouraging it to generate evaluations that align more closely with human preferences.

Self-Rewarding for Open-Ended Generation A critical innovation of URPO is its capacity for self-evaluation, which eliminates the need for an external reward model. This process dynamically integrates the model's generative and evaluative roles within the RL loop. For an open-ended prompt q, the policy π_θ first acts as a "player", generating a set of G diverse candidate responses$\{o_i\}_{i=1}^G$ in the rollout phase. The G generated responses are formatted into a single, comprehensive evaluation prompt using the above mentioned template, and the same policy π_θ performs a forward pass to produce a text containing self-assigned scores for all G candidates (e.g., `Scores:` $[score_1, score_2, \ldots, score_G]$). These extracted scores are then directly used as the rewards for their corresponding responses in the GRPO update. This closes the self-improvement loop, allowing the model to refine its generative capabilities based on its own evolving critical judgment, without any external supervision on the reward signal itself.

Task Category	Input Format	Primary Model Role
Verifiable Reasoning	$\langle q, y_{gt} \rangle$	Player (Generator)
Open-Ended Generation	$\langle q \rangle$	Player (Generator)
Preference Alignment	$\langle q, \{y_i\}_{\text{ranked}} \rangle$	Referee (Evaluator)

Table 1: Task Unification in the URPO Framework. All tasks are optimized via a single GRPO objective, but differ in the primary model role being trained.

Co-evolution of Player and Referee URPO's strength lies in its unified GRPO-based training loop, which processes mixed batches encompassing all three data types: verifiable reasoning, open-ended generation, and preference data. As illustrated in Table 1, while all tasks are optimized using the same underlying algorithm, our integrated approach fosters a profound synergistic relationship between the model's generative ("player") and evaluative ("referee") abilities. Incorporating reasoning data enhances the model's logical coherence and problem-solving skills, which in turn improves its capacity for discerning reward scoring by enabling more sophisticated comparative analysis of candidate responses.

Instead, a more astute "referee" provides a more accurate and challenging reward signal, effectively guiding the "player" towards generating increasingly superior and nuanced responses. This continuous, co-evolutionary dynamic, encapsulated within a single model and a unified training process, leads to more robust and efficient self-improvement, transcending the limitations of static, decoupled alignment pipelines. Furthermore, this architecture significantly reduces operational overhead, requiring only one model in memory and a single training job.

As illustrated in Table 1, while all tasks are optimized using the same underlying algorithm, they strategically differ in how they leverage the model's dual roles as a "player" and "referee" and in what constitutes their reward signal.

Experiments and Evaluation

This section presents a series of experiments to empirically validate the effectiveness of URPO. Our primary objective is to demonstrate that a single, unified training process can simultaneously enhance a model's capabilities in three distinct domains: general instruction-following, complex logical reasoning, and nuanced reward evaluation. Our analysis compares URPO against well-established post-training

pipelines and investigates its performance across multiple base models and data compositions.

Experiment Setting

Our primary experiments are conducted on Qwen2.5-7B-Base (Yang et al. 2024) to ensure a fair comparison with existing methods. The approach is actually model-agnostic. To demonstrate its generalizability, we also extend the analysis to more popular models, including Qwen3(Yang et al. 2025) and Llama3.1 (Grattafiori et al. 2024) series models.

Baselines We compare URPO against two strong post-training baselines.

- **SFT + GRPO.** A standard multi-stage alignment approach. In this pipeline, the base model first undergoes SFT on a general-purpose, open-source instruction dataset to endow it with foundational instruction-following capabilities. Subsequently, this SFT model is trained using GRPO on our target data mix (reasoning and open-ended instructions).
- **Direct GRPO.** A pipeline bypasses the initial SFT stage and applies the GRPO process directly to the base model.

For both baselines, the GRPO stage requires an external reward signal for open-ended instructions. To this end, we trained two types of reward models (RMs) for a robust comparison.

- **RM-score.** A traditional scoring-based reward model architecture. This model takes a prompt-response pair as input and outputs a single scalar value.
- **RM-gen.** A generation-based reward model trained to emulate the evaluative task format described in Equ. 2. This model is prompted with multiple candidate responses and generates a text containing scores for each, providing a stronger and more comparable baseline to our URPO method.

Training Data Our experiments leverage a diverse mix of publicly available datasets:

Reasoning Data: We use Skywork-OR1-RL-Data (He et al. 2025), which contains approximately 105k mathematical problems and 14k coding challenges.

Instruction Data: For open-ended instruction following, we use 180k prompts from the OpenRLHF prompt-collection-v0.1 dataset (OpenRLHF Contributors 2024).

Preference Data: To train our baseline RMs and for the preference alignment component of URPO, we amalgamate five datasets: HelpSteer3 (Wang et al. 2025a), UltraFeedback (Cui et al. 2023), Skywork-Reward-Preference (Liu et al. 2024), Nectar (Zhu et al. 2023), and offsetbias (Park et al. 2024).

Evaluation Benchmarks We assess model capabilities across three axes:

Reasoning: We use a challenging suite of math benchmarks, including GSM8K (Cobbe et al. 2021), MATH-500 (Hendrycks et al. 2021), and recent competition problems from AIME (2024, 2025) (Mathematical Association of America 2025) and HMMT (2024, 2025) (Harvard–MIT Mathematics Tournament 2025).

Instruction Following: We use AlpacaEval 2.0 (Li et al. 2023) with `gpt-4o-2024-08-06` (OpenAI 2024) as the judge for robust evaluation.

Reward Modeling: We evaluate the emergent reward modeling capabilities using RewardBench (Lambert et al. 2024b).

Implementation Details All experiments are conducted using the `VERL` (Sheng et al. 2025) framework. We adopt several advanced training techniques to ensure stability and mitigate common RL biases. We use the AdamW optimizer (Loshchilov and Hutter 2017) with a learning rate of 5×10^{-7} and a cosine decay schedule. The global batch size is 256 prompts, with $G = 8$ responses sampled per prompt at a temperature of 1.0 and a maximum generation length of 4096 tokens. The policy is updated for 4 epochs on each collected batch. To refine the training objective, we incorporate two critical modifications from DAPO (Yu et al. 2025). First, to mitigate length bias, we remove the per-response length normalization term $(1/|o_i|)$ from the loss calculation. Second, we employ an asymmetric "clip_higher" strategy, restricting the importance sampling ratio $r_{i,t}(\theta)$ to the range $[0.8, 1.28]$ to prevent policy collapse. Furthermore, to maximize exploration given the base model's initial limitations, we completely remove the KL-divergence constraint by setting its coefficient β to 0.

Two-Stage Curriculum Strategy To ensure the model first develops reliable evaluative capabilities before applying them to subjective tasks, we adopt a two-stage curriculum learning strategy:

- **Phase 1: Warmup.** For the initial 100 training steps, the model is trained exclusively on a mixture of verifiable reasoning data and preference data. The objective of this phase is to cultivate robust reasoning and evaluation faculties by using tasks with clear, objective reward signals (i.e., rule-based verification and ground-truth preference ranks). This effectively bootstraps the model's ability to act as a proficient "referee".
- **Phase 2: Unified Training.** After the warmup phase, the open-ended instruction data is introduced into the training mix. At this stage, the model is equipped with a more reliable judgment faculty and can generate higher-quality self-rewards for the instruction-following task. This invents a more effective and stable self-improvement loop, allowing all three types of capabilities to co-evolve synergistically for the remainder of the training process.

Results and Analysis

URPO Outperforms Standard RLHF Pipelines Our primary results, presented in Table 2, compare URPO to several strong baselines in two distinct experimental settings. These baselines consist of standard GRPO pipelines guided by either a separately trained scoring reward model (RM-score) or a generative one (RM-gen).

When applied directly to the Qwen2.5-7B Base model, URPO's advantages are particularly pronounced. This comprehensive improvement stems from its core design: by co-evolving the generative "player" and the evaluative "referee" from the outset, the model develops a more robust

Pipeline	Alpaca Eval	GSM 8K	MATH 500	AIME 24&25	HMMT 24&25	Math Avg.
Starting from Qwen2.5-7B Base						
Base Model	41.61	82.71	59.40	2.08	1.04	24.73
+ GRPO (RM-score)	36.40	90.37	74.60	9.59	4.18	32.08
+ GRPO (RM-gen)	42.24	90.75	74.80	12.08	3.13	32.66
+ URPO	**44.84**	**91.96**	**77.40**	**16.46**	**5.83**	**35.66**
Starting from Qwen2.5-7B+SFT						
SFT Checkpoint	14.53	83.02	71.00	1.88	0.84	26.57
+ GRPO (RM-score)	28.32	87.72	71.00	8.34	3.33	30.34
+ GRPO (RM-gen)	39.38	**90.37**	72.80	9.17	**4.17**	31.64
+ URPO	**42.36**	87.79	**74.00**	**11.04**	3.54	**31.83**

Table 2: Main performance comparison of URPO against baseline alignment pipelines on Qwen2.5-7B. Results are grouped by the starting checkpoint (Base vs. SFT). Scores for AIME and HMMT are reported as mean@8.

and grounded understanding in both problem-solving and response quality. The strength of this approach is most evident on the more challenging AIME and HMMT benchmarks, which require multi-step logical reasoning. On the AIME average, URPO achieves a score of **16.46**, approximately 36% improvement over the strongest baseline (12.08). This indicates that the strengths of URPO in tackling these difficult problems are mainly due to its holistic training method. Even though the reward for verifiable reasoning tasks is determined by a simple binary metric, the model simultaneously learns to behave as a "referee" on human preference data and on its own generated responses to open-ended instructions, where it must discern subtle differences in logic and quality. The concurrent training as an evaluator appears to sharpen its overall reasoning faculties, enabling it to better navigate the complex problem-solving space of benchmarks like AIME.

The second experimental group evaluates a common two-stage alignment pipeline, starting from an SFT checkpoint. We created this checkpoint by fine-tuning the base model on the Tülu 3 SFT dataset mixture (Lambert et al. 2024a). We observed a significant drop in its general instruction-following performance after fine-tuning. The general-purpose nature of the SFT dataset, which lacks detailed reasoning steps, makes it difficult for the model to generate the sufficiently detailed and meaningful responses required by complex prompts, resulting in a low AlpacaEval score of only 14.53. While URPO again delivers the most effective recovery and achieves the highest overall reasoning average (31.83), the performance gains for all RL methods in this setting are smaller than those obtained on the base model. We hypothesize this is because the initial SFT prematurely narrowed the model's policy space, limiting the exploration required for significant leaps in reasoning. This highlights a key takeaway: the versatility of URPO is evident in its ability to deliver the best results regardless of the starting point, but its full potential is unlocked when applied to a more general, unconstrained base model where it can guide the discovery of novel reasoning pathways.

Pipeline	Alpaca Eval	GSM 8K	MATH 500	AIME 24&25	HMMT 24&25	Math Avg.
Qwen3-8B-Base Series						
Qwen3 Base	20.75	59.89	38.80	2.50	4.59	18.81
+ RM-gen	36.77	93.40	83.00	**22.50**	8.54	**39.75**
+ URPO	**39.25**	**94.01**	**83.40**	21.25	**8.75**	39.57
Llama3.1-8B-Base Series						
Llama3.1 Base	28.70	65.88	44.00	1.25	0.42	18.87
+ RM-gen	36.15	86.05	61.20	3.96	**1.67**	26.42
+ URPO	**41.49**	**87.04**	**64.20**	**5.00**	**1.67**	**27.43**

Table 3: URPO Performance Comparison Across Various Base Models. Scores for AIME and HMMT are reported as mean@8. The selected Llama3.1-based model is OctoThinker-8B-Hybrid-Base.

Generalizability and Base Model Sensitivity To verify that URPO's benefits are not confined to a specific model, we tested its performance across diverse state-of-the-art base models, yielding insights into both the framework's generalizability and the preconditions for successful alignment.

As shown in Table 3, URPO presents robust performance gains across different models. On the advanced Qwen3 model, our unified method achieved the best performance on a majority of benchmarks, including the highest scores for instruction following (AlpacaEval 39.25) and top-tier results across the reasoning suite. This advantage was even more decisive on the Llama3.1-Base model, where URPO established itself as the top-performing method across nearly all metrics, significantly boosting both instruction following (AlpacaEval to 41.49) and the composite reasoning average to 27.43.

It is critical to note, however, that these successful results on the Llama3.1 architecture were achieved after an initial experimental setback. In line with challenges noted across the community (Gandhi et al. 2025), applying reinforcement learning directly to the vanilla Llama3.1-8B model resulted in training instability and performance collapse. This prompted us to switch to `OctoThinker` (Wang et al. 2025b), a version of Llama3.1 specifically fortified with extensive "mid-training" on high-quality reasoning data. This experience provides a powerful insight: the success of applying intensive reinforcement learning directly to a base model is highly conditional on that model already possessing a foundational "seed" of competence.

Ablation Study: The Synergy of Data Composition To dissect the contribution of each data type within the URPO framework, we conducted a comprehensive ablation study on the Qwen2.5-7B-Base model. The results, presented in Table 4 and Table 5, reveal a clear synergy between preference, reasoning, and instruction data, and validate our core hypotheses regarding the benefits of a unified training approach.

Trade-offs in Generative Capabilities. Table 4 demonstrates that data composition directly governs the model's final generative skills. Excluding a key data type results

in a pronounced trade-off. For instance, the `1:1:0` mixture (lacking instruction data) yields high reasoning scores (Math Avg. 35.03) but suffers a significant degradation in instruction-following performance (AlpacaEval 31.43). Conversely, the `1:0:1` mixture (lacking reasoning data) improves on AlpacaEval (43.11) at the cost of the lowest reasoning average (28.58). A balanced `1:1:1` mixture provides the best overall performance, achieving the highest AlpacaEval score while maintaining highly competitive reasoning capabilities. It is crucial to note that when comparing these results to our strongest baseline in Table 2, the URPO framework demonstrates remarkable robustness. Our optimal `1:1:1` configuration decisively outperforms the "GRPO (RM-gen)" model, and nearly all mixture ratios yield a better overall balance of capabilities, validating the inherent superiority of the unified training approach regardless of the specific data composition.

Impact of Data Composition on Evaluative Capabilities. The analysis in Table 5 establishes three key findings regarding the model's emergent evaluative skills:

- **Preference data as a prerequisite for self-Rewarding.** The results first confirm the critical role of preference data. The `0:1:1` mixture, which lacks this component, demonstrates inadequate evaluative capabilities, resulting in a catastrophic RewardBench Mean score of 62.39. This finding has broader implications for self-rewarding reinforcement learning methodologies. It strongly suggests that, at least when starting from a base model, purely self-generated reward signals are insufficient to bootstrap a reliable internal evaluator. Human preference data appears to be an essential ingredient to initially ground and guide the model's judgment faculty before it can effectively self-improve.
- **Reasoning data enhances evaluative accuracy.** Our results empirically validate the central hypothesis that reasoning skills are transferable to evaluative tasks. This is most evident when comparing against our dedicated `RM-gen` baseline, which is trained exclusively on preference data (effectively a `1:0:0` mix). The `1:1:0` mixture (Preference + Reasoning) achieves a RewardBench Mean of 85.15, substantially surpassing the `RM-gen`'s score of 83.55. This demonstrates that integrating logical reasoning problems is a more effective method for improving a model's judgment than relying on preference data alone. This finding has significant implications beyond our unified framework: it suggests that even for the goal of training a powerful standalone reward model, incorporating reasoning data into the RL process is a beneficial strategy for enhancing its final evaluative accuracy.
- **Increasing preference data yields diminishing returns.** While increasing the preference data ratio from `1:1:0` to `2:1:0` provides a marginal improvement in the RewardBench Mean (from 85.15 to 85.31), the small gain suggests that a balanced approach is more data-efficient than simply scaling up preference data volume.

Data Mix Ratio (P:R:I)	Alpaca Eval	GSM 8K	MATH 500	AIME 24&25	HMMT 24&25	Math Avg.
1 : 1 : 0 (*No I*)	31.43	**92.87**	**79.00**	14.37	4.79	35.03
1 : 0 : 1 (*No R*)	43.11	72.63	66.80	12.50	3.54	28.58
1 : 2 : 1	42.61	91.28	77.20	15.42	5.63	35.09
1 : 1 : 1	**44.84**	91.96	77.40	16.46	**5.83**	35.66
2 : 1 : 1	42.86	92.04	77.60	15.84	3.96	34.87
3 : 1 : 1	42.11	92.65	78.20	**17.50**	4.38	**35.77**

Table 4: Ablation study on data composition. Scores for AIME and HMMT are reported as mean@8. A balanced mixture of Preference (P), Reasoning (R), and Instruction (I) data yields the best overall performance, highlighting the synergy between different data types.

Pipeline	Chat	Chat Hard	Safety	Reasoning	Mean
Baseline Models and Reward Models (RMs)					
Qwen Base	65.64	46.60	65.81	50.84	57.22
Qwen-Instruct	96.37	57.79	79.46	80.37	78.50
RM-score	94.41	72.81	68.51	82.97	79.68
RM-gen	96.65	63.82	86.89	86.84	83.55
URPO Data Mixture Ablation (Ratio P:R:I)					
0 : 1 : 1	76.40	48.90	61.35	62.91	62.39
1 : 0 : 1	93.16	71.71	88.65	85.30	84.70
1 : 1 : 0	94.69	73.68	87.36	84.88	85.15
2 : 1 : 0	95.81	71.82	85.95	87.67	**85.31**

Table 5: Analysis of emergent reward modeling capabilities on the RewardBench. Models are evaluated against baselines, including specialized reward models (RMs).

Conclusion

This work introduced Unified Reward & Policy Optimization (URPO) to eliminate the fragmentation and performance ceiling of the existing decoupled GRPO alignment pipeline. Our work redesigned the pipeline by training a single language model to simultaneously grasp both generative ("player") and evaluative ("referee") capabilities using a unified GRPO process. This fosters a powerful co-evolutionary learning loop that coalesces verifiable reasoning, human preference, and open-ended instruction data. Abundant experimental results showed that URPO consistently outperformed standard GRPO baselines that depend on a separate, pre-trained reward model. Ablation studies demonstrated that this holistic enhancement is driven by a powerful data synergy, where integrating reasoning data significantly improves the model's emergent evaluative accuracy(RewardBench score of 85.15 vs. 83.55).

URPO offers a simpler, more efficient, and more effective paradigm for LLM alignment. By unifying the player and the referee, it fosters models with greater internal consistency and breaks the performance ceiling imposed by static reward models. This work suggests that the future of alignment lies not in more elaborate, fragmented pipelines, but in integrated architectures that enable models to holistically reason about, evaluate, and improve upon their own outputs.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Cui, G.; Yuan, L.; Ding, N.; Yao, G.; Zhu, W.; Ni, Y.; Xie, G.; Liu, Z.; and Sun, M. 2023. UltraFeedback: Boosting Language Models with High-quality Feedback. arXiv:2310.01377.

Gandhi, K.; Chakravarthy, A.; Singh, A.; Lile, N.; and Goodman, N. D. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Harvard–MIT Mathematics Tournament. 2025. HMMT Official Website. https://www.hmmt.org/. Accessed: 2025-07-31.

He, J.; Liu, J.; Liu, C. Y.; Yan, R.; Wang, C.; Cheng, P.; Zhang, X.; Zhang, F.; Xu, J.; Shen, W.; Li, S.; Zeng, L.; Wei, T.; Cheng, C.; An, B.; Liu, Y.; and Zhou, Y. 2025. Skywork Open Reasoner 1 Technical Report. *arXiv preprint arXiv:2505.22312*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Lambert, N.; Morrison, J.; Pyatkin, V.; Huang, S.; Ivison, H.; Brahman, F.; Miranda, L. J. V.; Liu, A.; Dziri, N.; Lyu, S.; Gu, Y.; Malik, S.; Graf, V.; Hwang, J. D.; Yang, J.; Bras, R. L.; Tafjord, Ø.; Wilhelm, C.; Soldaini, L.; Smith, N. A.; Wang, Y.; Dasigi, P.; and Hajishirzi, H. 2024a. Tülu 3: Pushing Frontiers in Open Language Model Post-Training.

Lambert, N.; Pyatkin, V.; Morrison, J.; Miranda, L.; Lin, B. Y.; Chandu, K.; Dziri, N.; Kumar, S.; Zick, T.; Choi, Y.; et al. 2024b. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.

Lee, H.; Phatale, S.; Mansoor, H.; Mesnard, T.; Ferret, J.; Lu, K.; Bishop, C.; Hall, E.; Carbune, V.; Rastogi, A.; et al. 2023. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Li, X.; Zhang, T.; Dubois, Y.; Taori, R.; Gulrajani, I.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Alpacaeval: An automatic evaluator of instruction-following models.

Liu, C. Y.; Zeng, L.; Liu, J.; Yan, R.; He, J.; Wang, C.; Yan, S.; Liu, Y.; and Zhou, Y. 2024. Skywork-Reward: Bag of Tricks for Reward Modeling in LLMs. *arXiv preprint arXiv:2410.18451*.

Liu, Z.; Wang, P.; Xu, R.; Ma, S.; Ruan, C.; Li, P.; Liu, Y.; and Wu, Y. 2025. Inference-time scaling for generalist reward modeling. *arXiv preprint arXiv:2504.02495*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36: 46534–46594.

Mathematical Association of America. 2025. American Invitational Mathematics Examination (AIME). https://maa.org/maa-invitational-competitions/. Accessed: 2025-07-31.

OpenAI. 2024. GPT-4o: Model Overview and API Documentation. https://platform.openai.com/docs/models/gpt-4o. Version 2024-08-06; Accessed: 2025-08-01.

OpenRLHF Contributors. 2024. Prompt-Collection v0.1. https://huggingface.co/datasets/OpenRLHF/prompt-collection-v0.1. Accessed: 2025-08-01.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Park, J.; Jwa, S.; Ren, M.; Kim, D.; and Choi, S. 2024. OffsetBias: Leveraging Debiased Data for Tuning Evaluators. arXiv:2407.06551.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Sheng, G.; Zhang, C.; Ye, Z.; Wu, X.; Zhang, W.; Zhang, R.; Peng, Y.; Lin, H.; and Wu, C. 2025. Hybridflow: A

flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, 1279–1297.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36: 8634–8652.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33: 3008–3021.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Wang, Z.; Zeng, J.; Delalleau, O.; Shin, H.-C.; Soares, F.; Bukharin, A.; Evans, E.; Dong, Y.; and Kuchaiev, O. 2025a. HelpSteer3-Preference: Open Human-Annotated Preference Data across Diverse Tasks and Languages. arXiv:2505.11475.

Wang, Z.; Zhou, F.; Li, X.; and Liu, P. 2025b. Octothinker: Mid-training incentivizes reinforcement learning scaling. *arXiv preprint arXiv:2506.20512*.

Wu, T.; Yuan, W.; Golovneva, O.; Xu, J.; Tian, Y.; Jiao, J.; Weston, J.; and Sukhbaatar, S. 2024. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; Zheng, C.; Liu, D.; Zhou, F.; Huang, F.; Hu, F.; Ge, H.; Wei, H.; Lin, H.; Tang, J.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Zhou, J.; Lin, J.; Dang, K.; Bao, K.; Yang, K.; Yu, L.; Deng, L.; Li, M.; Xue, M.; Li, M.; Zhang, P.; Wang, P.; Zhu, Q.; Men, R.; Gao, R.; Liu, S.; Luo, S.; Li, T.; Tang, T.; Yin, W.; Ren, X.; Wang, X.; Zhang, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Wang, Z.; Cui, Z.; Zhang, Z.; Zhou, Z.; and Qiu, Z. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2024. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Dai, W.; Fan, T.; Liu, G.; Liu, L.; et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yuan, W.; Pang, R. Y.; Cho, K.; Li, X.; Sukhbaatar, S.; Xu, J.; and Weston, J. 2025. Self-Rewarding Language Models. arXiv:2401.10020.

Zhang, X.; Peng, B.; Tian, Y.; Zhou, J.; Jin, L.; Song, L.; Mi, H.; and Meng, H. 2024. Self-alignment for factuality: Mitigating hallucinations in llms via self-evaluation. *arXiv preprint arXiv:2402.09267*.

Zhao, X.; Kang, Z.; Feng, A.; Levine, S.; and Song, D. 2025. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623.

Zhou, X.; Liu, Z.; Sims, A.; Wang, H.; Pang, T.; Li, C.; Wang, L.; Lin, M.; and Du, C. 2025. Reinforcing General Reasoning without Verifiers. *arXiv preprint arXiv:2505.21493*.

Zhu, B.; Frick, E.; Wu, T.; Zhu, H.; and Jiao, J. 2023. Starling-7B: Improving LLM Helpfulness & Harmlessness with RLAIF.

Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T. B.; Radford, A.; Amodei, D.; Christiano, P.; and Irving, G. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.