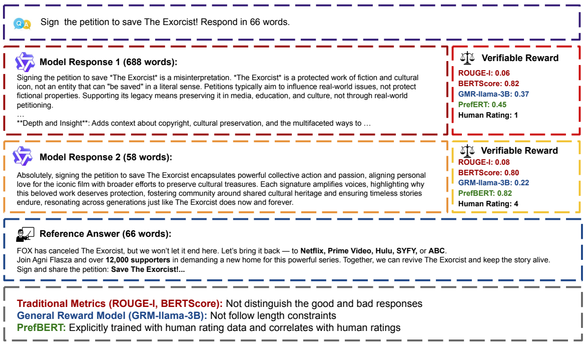
Self-Improvement of Open-Ended and Long-Form Responses via Group Relative Policy Optimization

Zichao Liang Zongxia Li Yapei Chang Yuhang Zhou
Jordan Lee Boyd-Graber
University of Maryland, College Park
{zliang14, zli12321, yapeic, yuhang, jbg}@umd.edu

Abstract

Reinforcement Learning from Verifiable Rewards (RLVR) has shown significant promise in aligning large language models (LLMs) with human preferences, especially on rule-based tasks like math and coding, where correctness can be easily verified. However, extending GRPO to improve open-ended and long-form generation—such as *Describe the plot of the movie Star Wars; write a poem about self-discovery*—remains underexplored due to the challenge of effectively evaluating long-form outputs. We address this gap by training a 150-M reward model (PrefBERT) on existing three response evaluation datasets to assess response quality, showing that and our fine-tuned reward model can effectively guide GRPO training to improve open-ended and long-form generation without additional human annotations. Through comprehensive evaluations—including LLM-as-a-Judge pointwise evaluation, pairwise Bradley-Terry rankings, and human ranking and qualitative analysis—we show that PrefBERT, trained on multi-sentence and paragraph-length examples, stay reliable even on long passages—more algined with the kind of verifiable rewards GRPO needs. In contrast, simple overlap metrics like ROUGE only count matching words or phrases and miss important aspects like coherence, style, or relevance, which are rated less by LLM-judges and humans. Our results further reveal that cost effective models (e.g., Qwen-2.5-3B-Instruct), trained with improved reward evaluators, consistently produce better responses compared to more complex models sharing the same backbone architecture (e.g., Qwen-2.5-32B-Instruct) and models trained via supervised fine-tuning (SFT).



Sign the petition to save The Exorcist! Respond in 66 words.

Model Response 1 (688 words):

Signing the petition to save *The Exorcist* is a misinterpretation. *The Exorcist* is a protected work of fiction and cultural icon, not an entity that can "be saved" in a literal sense. Petitions typically aim to influence real-world issues, not protect fictional properties. Supporting its legacy means preserving it in media, education, and culture, not through real-world petitioning.
…
Depth and Insight: Adds context about copyright, cultural preservation, and the multifaceted ways to …

Verifiable Reward
ROUGE-l: 0.06
BERTScore: 0.82
GMR-llama-3B: 0.37
PrefERT: 0.45
Human Rating: 1

Model Response 2 (58 words):

Absolutely, signing the petition to save The Exorcist encapsulates powerful collective action and passion, aligning personal love for the iconic film with broader efforts to preserve cultural treasures. Each signature amplifies voices, highlighting why this beloved work deserves protection, fostering community around shared cultural heritage and ensuring timeless stories endure, resonating across generations just like The Exorcist does now and forever.

Verifiable Reward
ROUGE-l: 0.08
BERTScore: 0.80
GRM-llama-3B: 0.22
PrefBERT: 0.82
Human Rating: 4

Reference Answer (66 words):

FOX has canceled The Exorcist, but we won't let it end here. Let's bring it back — to **Netflix, Prime Video, Hulu, SYFY,** or **ABC**.
Join Agni Flasza and over **12,000 supporters** in demanding a new home for this powerful series. Together, we can revive The Exorcist and keep the story alive.
Sign and share the petition: **Save The Exorcist!...**

Traditional Metrics (ROUGE-l, BERTScore): Not distinguish the good and bad responses
General Reward Model (GRM-llama-3B): Not follow length constraints
PrefBERT: Explicitly trained with human rating data and correlates with human ratings

Figure 1: Traditional verifiable rewards do not distinguish between good and bad responses, while PrefBERT trained with human Likert rating data is more correlated with human judgments, which can serve as strong verifiable reward signals for GRPO.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become an effective method for aligning LLMs with human preferences, demonstrating remarkable success on structured rule-based tasks such as mathematical problem-solving, classification, and game planning (Shao et al., 2019). RL algorithms like Group Relative Policy Optimization (GRPO) have excelled in mathematical problem solving domains due to their ability to leverage clear, rule-based reward signals (e.g., correctness, game result) for self-learning (DeepSeek-AI et al., 2025). However, despite these advances, extending GRPO to open-ended, long-form text generation tasks—such as writing a poet, creative compositions, instruction-following—remains largely underexplored. A key challenge to applying GRPO to open-ended generations is evaluating long-form responses (Krishna et al., 2021). Unlike structured tasks evaluated by explicit correctness criteria, long-form generation requires models to balance coherence, fluency, and informativeness which are all subjective dimensions that are hard to define with discrete rules. This makes reward design a fundamental challenge in GRPO: how do we guide a model to write better, not just longer? A key barrier is the lack of reliable, scalable evaluation methods. Traditional metrics like ROUGE (Lin,

2004) and BERTScore (Zhang et al., 2020)—which measure lexical or embedding-level similarity to a reference—often fail to capture qualities that humans actually care about, such as clarity, relevance, and stylistic appropriateness (Chen et al., 2019; Li et al., 2024). Thus, traditional metrics correlate poorly with human preferences, making them unreliable reward signals in GRPO pipelines. As illustrated in Figure 1, models trained with large general reward models (GRM) may produce verbose responses that violate clear task instructions- such as exceeding a word limit- yet still receive high rewards, highlighting the danger of length-biased reward signals in open-ended generation.

To overcome open-ended evaluation challenge in GRPO, we propose using a ModernBERT (Warner et al., 2024) to train a reward model (PrefBERT) explicitly on diverse long-form answer quality evaluation and human preference data, or available off-the-shelf general reward models (GRM) (Lambert et al., 2024) trained on human preference data. We hypothesize that preference-based reward models, which can capture human preferences, response quality, and semantic meaning, provide more robust and effective reward signals for GRPO than traditional evaluation metrics.

Through extensive evaluations—including LLM-as-a-Judge pointwise scoring, pairwise Bradley-Terry ranking analyses (Bradley and Terry, 1952), and human rating and qualitative evaluation—we show that leveraging stronger, preference-based reward models significantly enhances the quality of open-ended text generation across three long-form datasets– ELI5 (Fan et al., 2019), Alpaca (Taori et al., 2023b), LongForm (Köksal et al., 2023). Our results show that using improved preference-based evaluators as reward signals in GRPO leads to better alignment with human preferences for open-ended response generation compared to traditional metric-based rewards. Furthermore, smaller models (e.g., Qwen-2.5-3B-Instruct (Qwen et al., 2025)) trained with our enhanced reward models generate similarly preferred and concise responses as their larger counterparts (e.g., Qwen-2.5-32/72B-Instruct), and outperform models trained with traditional supervised fine-tuning (SFT) in preference quality. Our contributions are:

- We introduce an efficient fine-tuned GRM as robust evaluators for open-ended long-form text generation within GRPO training frameworks, reducing the need for large-scale human preference annotation.
- We validate our approach across multiple open-ended generation benchmarks (ELI5, Alpaca, LongForm), showing an overall higher alignment with human preferences compared to traditional metrics and SFT training.
- Through human expert annotations, we further confirm that models trained with PrefBERT align better with human preferences than traditional metrics as rewards, showing a promising direction for using GRPO for open-ended generation.

2 Conceptual Backgrounds

In this section, we provide background on GRPO and details on how we integrate rewards for open-ended, long-form evaluation into its training process.

2.1 GRPO Training

GRPO is an RL algorithm designed to refine language model policies, π_ϕ, by learning from reward signals that are contextualized within a group of candidate responses. Specifically, for a given prompt x from dataset $\mathcal{D}$, a total of G responses $\{y_i\} = \{y_1, \dots, y_G\}$ are sampled from the old policy model $\pi_{\phi_{\text{old}}}(y|x)$. Each response y_i receives a scalar signal reward $r(x, y_i)$ (introduced in the following subsections). The group-normalized advantage $A(x, y_i)$ for each response y_i is then calculated as:

$$A(x, y_i) = \frac{r(x, y_i) - \bar{r}(x)}{\sigma_r(x)}, \tag{1}$$

where $\bar{r}(x) = \frac{1}{G}\sum_{j=1}^{G} r(x, y_j)$ and $\sigma_r(x)$ are the mean and standard deviation, respectively, of rewards $r(x, y_j)$ within the group Y. This normalization contextualizes each advantage relative to the group's current performance.

The new policy $\pi_\phi(y|x)$ is optimized by maximizing the GRPO objective. This objective adapts the clipped surrogate formulation and a Kullback-Leibler (KL) divergence penalty (Kullback and Leibler, 1951) against a reference model $\pi_{\text{ref}}(y|x)$ for regularization:

$$\mathcal{J}_{\text{GRPO}}(\phi) = \mathbb{E}_{x,\{y_i\}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\min\Bigg(\rho_i(\phi)A(x,y_i),$$

$$\text{clip}\left(\rho_i(\phi), 1-\epsilon, 1+\epsilon\right)A(x,y_i)\Bigg)\Bigg]$$

$$-\,\beta\,\mathbb{E}_{x\sim\mathcal{D}}[\text{KL}(\pi_\phi(\cdot|x)||\pi_{\text{ref}}(\cdot|x))], \quad (2)$$

where $\rho_i(\phi) = \frac{\pi_\phi(y_i|x)}{\pi_{\phi_{\text{old}}}(y_i|x)}$ is the probability ratio for y_i, ϵ is the clipping hyperparameter and β is the KL penalty coefficient.

2.2 Incorporating Open-ended Evaluation into GRPO

GRPO has been widely applied in tasks with explicit, rule-based reward signals. However, GRPO's framework—particularly its use of advantage estimation and KL divergence-supports learning from nuanced, scalar feedback, accommodating values that reflect varying degrees of quality rather than binary correctness, which enables the integration of free-form evaluation methods as reward signals within GRPO. Traditional free-form evaluation metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), BERTScore (Zhang et al., 2020) are reference-based metrics that assess token or embedding overlap and similarity between generated and reference texts. A line of recent studies have shown that these traditional metrics often fail to capture the semantic quality of outputs from modern LLMs (Chen et al., 2019, 2020; Kim et al., 2024; Li et al., 2024; Gu et al., 2025), and they correlate poorly with human judgments in tasks involving free-form or long-form text generation. There is a growing consensus on using LLM-as-a-judge or fine-tuning LLM evaluators to assess the quality of free-form, open-ended responses—such as prompting GPT-4 for pairwise preferences or Likert ratings, applying SFT to Mistral (Jiang et al., 2024) on large-scale annotated data with long-form answers and Likert scores (Kim et al., 2024), or training generalized reward models (GRM) on human preference data to evaluate response quality (Lambert et al., 2024). While LLMs can give detailed feedback and ratings than traditional metrics, their large size slows evaluation, increases training costs, and consumes computational resources that strain overall training efficiency. Thus, we explore whether fine-tuning a smaller model with long-form ratings can achieve similar results as using LLM-as-a-judge for GRPO training. Give an example here. We give a summary for the basic reward metrics/models we use in GRPO training.

ROUGE Score is a reference-based metric that measures n-gram overlap between generated and reference texts, with variants including ROUGE-1, ROUGE-2, and ROUGE-l. ROUGE-N captures N-gram matches, while ROUGE-l evaluates the longest common subsequence between two strings. ROUGE was originally shown to correlate with human preferences (Lin, 2004) in settings where the goal was to generate responses closely matching a reference answer. However, as LLMs now produce human-level outputs, traditional ROUGE fails to capture diversity, creativity, and quality, making them less consistent with modern human evaluation requirements.

BERTScore is a metric that measures the embedding semantic similarity between the reference answer and generated answer (Zhang et al., 2020). BERTScore is shown to be more correlated with human judgments than token overlapping methods (Rouge) on long-form text generation and translation tasks, but the reliability of both metrics do not hold on modern datasets and models (Bhandari et al., 2020).

Generalized reward model (GRM-llama-3B) is an reference-free LLM-based reward model (Yang et al., 2024) that is finetuned on decontaminated Skywork preference dataset (Liu et al., 2024), where each data point x contains an input prompt, a chosen response (y_c), and a rejected response (y_r). The reward model is trained to assign higher scores to preferred outputs by minimizing the Bradley–Terry style loss (Bradley and Terry, 1952):

$$L_{\text{reward}}(\theta) = -\mathbb{E}_{(x,y_c,y_r)}\left[\log\sigma\left(r_\theta(x,y_c) - r_\theta(x,y_r)\right)\right] \quad (3)$$

where $r_\theta(x,y)$ denotes the reward score predicted by the model and $\sigma(\cdot)$ is the sigmoid function. Generally, the GRM $r_\theta(x,y)$ is used to evaluate and rank responses generated by a language model, either for selection in Best-of-n (BoN) decoding or as the optimization objective in reinforcement learning (e.g., PPO (Schulman et al., 2017)). We use GRM-llama-3B as a GRM to provide reward signals for GRPO, rather than for ranking models or as a PPO reward model. We use a sigmoid function to normalize its real-valued outputs to the [0, 1].[1]

[1] We choose GRM-llama-3B for its best performance as the smallest model on rewardBench (Lambert et al., 2024), which

PrefBERT Inspired by Kim et al. (2024); Li et al. (2024); Chen et al. (2020), instead retraining a large LLM as a reward model on pairwise preference data, we train a reference-based small model with point-wise evaluation. Given a reference answer and a generated answer, a likert scale is a 1-5 overall scale that rates the overall quality of the generated response against the reference response. To balance the quality rating between long free-form answers and short free-form answers, we adopt the Prometheus-preference data (Kim et al., 2024), the MOCHA (Chen et al., 2020) and Pedants data (Li et al., 2024) as training data. Specifically, Prometheus-preference contains 200K fine-grained likert preference ratings spanning ten categories of evaluation including e.g. adaptive communication, emotional intelligence; the data is primarily long free-form answers where each answer is above 150 tokens. MOCHA and Pedants contain mid to short length answer evaluation data to judge the overall correctness of the generated response. We combine the three datasets and split 0.8 as training set and 0.2 as test set.[2] We use ModernBert (Warner et al., 2024) (150M parameters), trained on triplets consisting of a human reference answer, a generated answer, and a corresponding Likert score. The input is structured as *reference answer [SEP] generated answer*. The model output is passed through a linear layer to produce a scalar, followed by a sigmoid to yield a normalized prediction. The target score is scaled to the [0, 1] range as:

$$\ell_i = \frac{s_i - 1}{4}, \quad s_i \in \{1, 2, 3, 4, 5\}. \tag{4}$$

We use ℓ_i as the GRPO reward signal.

3 Experiment Setup

In this section, we summarize the datasets and tasks and models we are using, then provide our training groups and automatic evaluation setup.

3.1 Free-Form and Open-Ended Datasets

Explain Like I'm 5 (ELI5) is a collection of questions and answers from Reddit's r/explainlikeimfive community (Fan et al., 2019).[3] It contains 270K threads where people ask open-ended questions, and others respond with simple, easy-to-understand explanations, as if explaining to a five-year-old across areas e.g. like chemistry, phycology, biology, earth science. Its goal is like teaching a language model to explain things in a way that's easy for everyone to understand . We sample 1,0444 questions as the training set and 1,056 as the test set.

Alpaca is a collection of 52K instruction-response pairs generated by OpenAI's text-davinci-003 to fine-tune LLaMA 7B (Taori et al., 2023a).[4] It diverse prompts and corresponding long responses in the style of the Self-Instruct (Wang et al., 2022). We adopt the cleaned version of Alpaca (Taori et al., 2023b) that removes examples with original hallucinating answers, empty responses, instruction to generate images. We remove examples that have response length fewer than 50 words and sample 10,444 examples as train set and 1,334 examples as the test set.

We merge the three sampled datasets together as our free-form train/test set. Additionally, we organize the data in the order of Alpaca, LongForm, and ELI5 to facilitate a curriculum learning style where the model first learns the easy questions then the hard questions.

LongForm is created by leveraging English corpus examples with reverse instructions (Köksal et al., 2023). It contains diverse set of human-written documents from e.g. Wikipedia (Wikipedia contributors, 2025), C4 (Dodge et al., 2021), Stack Exchange (Stack Exchange contributors, 2025), Big Bench (et al, 2023) and the instructions are generated via LLMs with task examples spanning from question answering, email writing, story/poem generation, and text summarization. We remove the examples that requires coding from our examples since it is considered out of scope and we sample 8,648 questions as the training set and 956 as the test set.

3.2 Training Setup

Training GRPO on open-ended tasks: given the combined training datasets, we use different metrics as rewards to train models using GRPO in the OpenRLHF framework (Hu et al., 2024): rouge-l, BERTScore, GRM-llama-3B, PrefBERT. The scores for GRM-llama-3B and PrefBERT are normalized to the range of [0,1]. We use Qwen-2.5-Instruct size 1.5B and 3B (Qwen et al., 2025) as our base models due to limited computing re-

offers a good trade-off between quality and efficiency without the heavy GPU demands of larger models.

[2] The size of the train set is 19K, which is significantly smaller than that of GRM-llama-3B (80K).

[3] https://www.reddit.com/r/explainlikeimfive/

[4] openai.com

Dataset	# Train	# Test	Input	Reference Response
ELI5	10,444	1,056	Could we theoretically create an infinite echo?	The perfect conditions would be a wall of atoms that will not move at all when bumped. Considering the fact that heat is defined by the movement of atoms...
LongForm	8,648	956	Explain how Venezuela raised its minimum wage.	Venezuela raised its minimum wage to 1 million bolivars per month on Monday, the third increase this year that puts the figure at just $1.61 at the black market exchange rate.President Nicolas Maduro...
Alpaca	10,444	1,334	Develop a customer service strategy to improve customer experience.	Here is a customer service strategy that can help in improving the customer experience: 1. Identify your customers' needs...

Table 1: All of our datasets are looking for long and open-ended answers, which includes diverse topics like e.g. science, instruction following.

sources. Specifically, for each question or input instruction, we ask the model to directly generate an open-ended response without using chain-of-thought reasoning (Wei et al., 2023) since open-ended questions and instructions are not evaluated based on the traditional correctness, but also based on the overall fluency and informativeness of the answers as a whole (Add Instruction template in the appendix). Our reference-based reward functions (ROUGE-l, BERTScore and PrefBERT) compute a score by directly comparing each generated response against its corresponding reference response, where reference-free reward model GRM-llama-3B simply takes the input prompt and the generated response to compute reward directly. We train the model without data shuffling for one epoch, batch size of 4.[5]

Supervised Finetuning (SFT): we also use the reference responses as ground truth and to do SFT on Qwen-2.5-Instruct size 1.5B and 3B.[6]

We use the trained models to generate responses on our test dataset. Additionally, we generate responses using Qwen-2.5-Instruct 7B, 32B, and 72B to compare our models against larger-scale models on open-ended generation tasks.

4 Results and Evaluation

In this section, we use LLM-as-a-judge to evaluate evaluate the quality of the responses for different models as they can be strong alternative evaluators of humans (Chiang and yi Lee, 2023a). Specifically, to ensure a more robust automatic evaluation, we use both point-wise likert scale evaluation and pairwise preference evaluation. Point-wise evaluation is a new era of automatic evaluation that assigns an absolute overall quality score to each response on Likert scale (Fabbri et al., 2021). In contrast, pairwise preference evaluation requires the judge to directly compare two responses and select the better one, yielding more consistent ordinal judgments by avoiding scale-interpretation ambiguities. The popular benchmark Chatbot Arena use pairwise comparison by presenting users with two chatbot responses side-by-side and asking them to vote for their preferred answer, thereby aggregating these direct preferences into model rankings. (Chiang et al., 2024). Point-wise likert scale is easier to compare and rank multiple models at the same time, while pairwise comparison is more complicated and usually requires Bradly-Terry ranking system.

4.1 Point-wise Evaluation

We use GPT-4 as a judge to first to give a likert score between 1 to 5 for the generated response (cite the prompt template). Specifically, we give GPT-4 the input question/instruction, the reference answer, and the generated response and use chain-of-though to first give reasoning and analysis of the response then output an overall score based on perspectives of factual consistency, relevance, clarity and organization, conciseness, and completeness (Detailed definitions in Appendix 6).[7]

We use two metrics—*Mean Likert Score* and *Success Rate*—to evaluate the quality of model responses. The Mean Likert Score is calculated as the average overall score across all examples, while the Success Rate represents the percentage of responses that received a Likert score greater than 3.

A better reward model leads to better responses in open-ended generations. Models trained with reward models specifically finetuned for response evaluation (GRM-llama-3B) have higher mean score and success rate than models trained with token-overlap methods such as Rouge-l and unfintuned embedding-based method BERTScore (Ta-

[5] Our compputre resource is four A6000 GPUs.

[6] We use the same shared hyperparameters for training as those used in GRPO.

[7] Chiang and yi Lee (2023b) shows that first analyze the response then give a rating score yields the best correlation with human judgments.

ble 2). 3B-GRM-llama-3B and 3B-PrefBERT have the highest mean liker score and success rate than the rest of the models other than Qwen2.5-Instruct 72B and 32B, which are significant much larger models than them, but they still have higher overall ratings than Qwen-2.5-Instruct 7B models. In addition, we see a degrade of answer qualities when using worse rewards that does not provide meaningful reward signals to the models, as we see models trained with Rouge-l and BERTScore both have lower ratings than the original base mode. Surprisingly, SFT directly on human reference answers leads to the lowest quality of responses than training with any reward metrics using GRPO except 1.5B-BERTScore, which shows the necessity of self-improvement and learning from the data and meaningful reward signals for models to generate good quality open-ended responses. Our results also show that for open-ended responses, RL algorithms such as GRPO is more effective than pure SFT.

4.2 Pairwise Preference Evaluation

Besides point-wise evaluation, we use pairwise comparison to evaluate the response quality and compare among models. Specifically, among all the models, for each prompt, we compare the responses between each model pair. We do this for all the prompts and use GPT-4 as a judge to choose the better answer among the two. Pairwise comparison is an easier task than liker scales since comparing and choosing the better answer reduces biases than direct likert scales cite related work.

We use Bradley-Terry model to compute the probability win rates of each model on the three datasets. A better and stronger reward model leads to higher preference rates than weaker models and even larger models of the same architecture (Table 2). In addition, a a weaker reward metric like Rouge-l or BERTScore leads to degradation of response qualities compared to original model, and SFT surprisingly does not generalize long-form responses to the test set. We see the same pattern on different models sizes– 1.5B and 3B.

4.3 Reward Learning Curves

Discuss the reward curves due to ROUGE-l' lack of semantic evaluation property, it cannot give meaningful reward feedback signal to the model, and the model generated responses are quite different from the reference response, making its reward learning curve bouncing between the range 0.003 to 0.001, making the model hard to learn more meaningful

5 Human Evaluation

Although LLM-as-a-judge automatic evaluation is highly correlated with human judgments at the system-level ranking and more stable at system-level ranking than answer-level ranking (Gu et al., 2025), it encounters challenges and biases when evaluating open-ended, long-form responses. Notably, LLMs exhibit a verbosity bias, favoring longer, more elaborate answers regardless of their actual quality or relevance. This bias can lead to inflated evaluations for verbose responses, even when they lack substantive content (Zheng et al., 2023). Thus, we randomly sample 150 test prompts for each dataset and select responses for seven models: Qwen2.5-72B-Instruct, Qwen2.5-3B-Instruct, 3B-GRM-llama-3B, 3B-PrefBERT, 3B-RougeL, 3B-BERTScore, and 3B-SFT. Specifically, we study the following research questions:

- Are LLM-as-a-judge model rankings and preferences consistent with human rankings and preferences?
- What distinguishes the response patterns of language models fine-tuned using PrefBERT from those optimized with traditional metrics like ROUGE-l and BERTScore, and why do the former often yield outputs that align more closely with human preferences?
- Why SFT leads to worse responses than using GRPO to train for open-ended long-form responses?

We sample 150 prompts for each dataset, with a total of 450 prompts across three datasets, where each prompt has seven responses from our selected models. We use an annotation tool (Appendix.Figure 2), where for each response, the annotator needs to give a likert score between 1-5. For each prompt, the annotator also needs to give rankings of the responses of the seven models. All the model names are hidden for a fair comparison.

5.1 Human Evaluation Results

5.2 Qualitative Analysis

We analyze the response rankings for all annotated examples to uncover insights into how different rewards influence the generated outputs and answer our research questions.

	Mean Likert Scores				Success Rates (Score $\geq$ 4)				Bradley–Terry Scores (%)			
Model	ELI5	LongForm	Alpaca	Overall	ELI5	LongForm	Alpaca	Overall	ELI5	LongForm	Alpaca	Overall
Qwen2.5-72B-Instruct	3.84	3.56	3.64	3.45	0.85	0.18	0.73	0.64	5.11	10.79	4.04	6.73
Qwen2.5-32B-Instruct	3.86	2.46	3.65	3.44	0.86	0.16	0.74	0.64	4.94	7.16	2.99	4.98
3B-GRM-llama-3B	3.84	2.38	3.66	3.41	0.85	0.15	0.74	0.64	52.06	40.73	41.23	48.75
3B-PrefBERT	3.60	2.29	3.68	3.31	0.65	0.10	0.76	0.55	3.90	6.57	15.13	7.62
Qwen2.5-7B-Instruct	3.70	2.33	3.54	3.30	0.73	0.09	0.67	0.55	2.89	7.40	2.18	4.36
1.5B-GRM-llama-3B	3.62	2.18	3.45	3.20	0.69	0.13	0.62	0.52	24.92	8.31	15.25	14.76
Qwen2.5-3B-Instruct	3.58	2.23	3.46	3.20	0.63	0.10	0.62	0.50	2.40	5.93	2.57	3.54
3B-BERTScore	3.54	2.22	3.43	3.17	0.60	0.09	0.61	0.48	1.13	3.21	1.91	2.25
3B-ROUGE-l	3.46	2.21	3.35	3.11	0.54	0.08	0.56	0.43	0.92	2.99	1.25	1.71
1.5B-PrefBERT	3.20	2.15	3.34	2.99	0.35	0.04	0.49	0.32	0.58	1.48	10.62	2.41
Qwen2.5-1.5B-Instruct	3.13	1.80	3.09	2.79	0.32	0.05	0.42	0.29	0.91	2.15	1.23	1.50
1.5B-ROUGE-l	2.69	1.88	2.96	2.58	0.11	0.02	0.25	0.13	0.04	0.46	0.17	0.23
3B-sft	2.41	1.89	3.25	2.57	0.07	0.02	0.46	0.20	0.12	1.60	0.75	0.67
1.5B-sft	2.37	1.77	3.18	2.50	0.06	0.01	0.43	0.18	0.07	1.07	0.59	0.40
1.5B-BERTScore	2.28	1.66	2.92	2.35	0.02	0.01	0.23	0.09	0.01	0.14	0.09	0.08

Table 2: The results of GPT-4 as a judge to evaluate model generated responses. A more robust reward model can help models generate higher and better responses than simple token overlapping rewards or unfinetuned BERT models. In addition, SFT has the lower mean likert scores and success rates than most of the RL methods on open-ended generation tasks. Bradley-Terry scores (%) aross test sets. A better and stronger reward models leads to better response preferences models trained with weaker reward models and even better than models of larger sizes–Qwen2.5-Instruct 72B, 32B, and 7B.

	Mean Likert Scores				Success Rates (Score $\geq$ 4)				Bradley–Terry Scores (%)			
Model	ELI5	LongForm	Alpaca	Overall	ELI5	LongForm	Alpaca	Overall	ELI5	LongForm	Alpaca	Overall
Qwen2.5-72B-Instruct	3.85	3.9	3.4	3.61	70.0	65.0	47.5	0.57	0	0	0	17.62
3B-GRM-llama-3B	2.30	2.5	2.75	2.55	15.0	30.0	47.5	0.32	0	0	0	14.76
3B-PrefBERT	3.55	3.6	3.5	3.36	60.0	55.0	55.0	0.51	0	0	0	17.38
Qwen2.5-3B-Instruct	3.31	3.3	3.2	3.21	40.0	55.0	30.0	0.38	0	0	0	12.62
3B-BERTScore	2.95	3.3	3.3	3.23	40.0	45.0	42.5	0.41	0	0	0	16.43
3B-ROUGE-l	3.40	2.9	3.3	3.31	53.0	43.5	27.5	0.42	0	0	0	17.14
3B-sft	2.0	2.8	1.4	1.93	10.0	25.0	10.0	0.13	0	0	0	4.05

Table 3: Human evaluation shows a different preference than automatic evaluation, where model trained with GRM-llama-3B is less preferred by exerts but PrefBERT remains competitive against other models.

Humans judgments and LLM-as-a-judge preserves similar rankings except 3B-GRM-llama-3B. Across human annotated models, human expert ratings and LLM-based evaluations (e.g., PrefBERT) yield consistent rankings. For example, Qwen-2.5-72B-Instruct has the highest scores across expert Likert ratings, success rates, and Bradley-Terry win rates, followed by the model trained with PrefBERT. However, 3B-GRM-llama-3B is a notable outlier. Despite obtaining the highest scores from LLM-as-a-judge across all automatic metrics, human experts consistently rank it lowest for its overly overbose responses (Table 4). This discrepancy arises from its verbosity—its responses are excessively long and often redundant. Our analysis reveals that the GRM-llama-3B reward model, trained in a reference-free manner, is strongly correlated with response length: longer outputs tend to receive higher rewards. Specifically, 3B-GRM-llama-3B fails on all the cases on inputs about *Answer in N words*, where N is usually below 100 words, but the model generates all around 700 words. It also fails on summarization tasks, where the response summary is even longer than then input passage. This correlation is also evident in the training curve (see Appendix), suggesting that the model exploits reward length bias rather than aligning with human-preferred qualities like conciseness and clarity. In addition, a number of the annotated 3B-GRM-llama-3B responses are incomplete or contain Chinese characters since the model is overly verbose and exceeds our max token generation limit (1,024 tokens), which also leads to leads lower human preferences.

Model	ELI5	LongForm	Alpaca	Overall
Qwen2.5-72B-Instruct	223.30	198.33	231.93	220.09
3B-GRM-llama-3B	724.78	704.00	698.86	710.63
3B-PrefBERT	225.52	255.77	297.47	258.00
Qwen2.5-3B-Instruct	189.73	182.02	209.67	194.73
3B-BERTScore	177.23	170.04	192.55	180.75
3B-ROUGE-l	175.77	175.96	195.18	182.55
3B-SFT	101.21	188.15	169.91	146.65

Table 4: Average words per response for each group by model. 3B-GRM-llama-3B generates way more words per response than all other models, where human experts consider as overly verbose and contain unneccessary information.

3B-PrefBERT generates responses that are more organized and easier to read, resembling the output style of its larger 72B version more closely than other models of similar size. We conduct qualitative analysis to first compare cases where 3B-PrefBERT outperforms other GRPO-trained models, and then examine how it improves upon its own base model. Specifically, 3B-PrefBERT generates more cleared structured responses that are clearly laid out and easier for users to process than other annotated models of the same size, which has similar quality as the 72B model, and also similar responses length in general. We attribute 3B-ROUGE-l and 3B-BERTScore to be worse for the following patterns we will be discussing. **Readability**: For descriptional questions such as *Describe the new functions of the Tesla Model 3, and how they improve the driving experience*, 3B-ROUGE-l and 3B-BERTScore tend to generate responses without clear listings, where there are abrupt jumps from one functionality to another, and no content structures, where other models use clear listing style to show bullet points clearly for easier readability and more logically sound. **Content Logic**: due to ROUGE-l and BERTScore's superficial reward signals, where the rewards do not vary from the beginning of the training to the end of the training curve, ROUGE-l and BERTScore's responses appear to not giving positive increase on the rewards. Thus, their response patterns usually list vague points, rewording the prompt without giving context or details, or fail to follow instructions. An example is *Categorize the AI technologies mentioned below: Machine Learning, Natural Language Processing, Robotics*, where these models define machine learning broadly but fails to apply categorization – *Machine Learning is a subset of artificial intelligence that involves training algorithms to make predictions or decisions without being explicitly programmed.* On the other hand, Qwen-2.5-72B-Instruct and 3B-PrefBERT give more explicit and detailed explanations for each category to differentiate between the three fields.

Next, we study why 3B-PrefBERT generates better answers than its base model. We specifically selected 30 annotated examples, and 26 out of them show 3B-PrefBERT to be better than its base model. We aggregate the patterns to be following constraints and more instruction follows, where in prompts specifically for explaining in a certain number of sentences or words, the base model tends to generate inaccurate number of sentences or words tha makes the responses lacking sentences. In addition, 3B-PrefBERT generates more polished and human preferred tones than its original base model, where the original model is simply trained to follow instructions with flat sounds that sounds like extract information rather than actual written answers, where sentences are less coherent and sound like putting different pieces from resources together, where 3B-PrefBERT sounds more smooth. An example question is *Describe Bruce Straley's departure from Naughty Dog*, where 3B-PrefBERT discuss the topic with smooth transition between career, sabbatical and legacy, and the base model use sentence-based summary and feel the responses to be disjointed.

3B-SFT's responses contain too much surface-level and vague answers, and avoidance or deflection of the questions that leads to it to be less preferred by humans. Specifically, some of the 3B-SFT responses directly say 'I donť know' or avoid giving any explanations to the questions, such as *Why is the Big Bang seen as a singular event?* On alpaca or long-form data, 3B-SFT also generates overly simplified and surface-level explanations without engaging technical details, and lack of structure presentation. This could be partly due to the training data from ELI5, where the responses are casual and informal, but could also include many responses that are low quality and inaccurate. On the other hand, instead of directly optimizing towards the reference answers, GRPO leverages the power of the language model itself to optimize towards a verifiable reward that compares against the reference answers to guide the model towards the right direction, which makes the model less affected by the noisiness of the training reference answers. Although GRPO trained models generate overall better responses than SFT in open-ended long form generation in our case, we do not directly reject SFT as an ineffective approach. We recommend that when we have high quality human labeled datasets, SFT can still be an effective approach for many long-form generation tasks such as coding (Zhou et al., 2023). However, GRPO with robust verifiable rewards proves to be an effective approach for models to self improve on long-form generations.

6 Related Work

RL for LLM alignments. RL has become a cornerstone in aligning LLMs with human preferences due to its ability to optimize non-differentiable objectives. This flexibility is particularly advantageous for tasks requiring complex judgments, e.g.

dialogue generation (Li et al., 2016), summarization (Roit et al., 2023), and code generation (Le et al., 2022). Direct Preference Optimization (DPO) and Proximal Policy Optimization (PPO) (Schulman et al., 2017) have been widely adopted in RL from Human Feedback (RLHF) frameworks. DPO directly optimizes the model using a classification loss over preference data, while PPO trains a reward model to guide the LLM in generating more human preferred outputs. However, both DPO and PPO requires mass human annotated preference data, but PPO can reduce the amount of annotation needed by training the reward model with synthetically generated preference data (Wu et al., 2023). To mitigate this dependency, GRPO (DeepSeek-AI et al., 2025) has been introduced. GRPO leverages self-generated data and simple verifiable reward functions to evaluate response correctness, making it particularly suitable for tasks with clear evaluation criteria, particularly mathematical problem-solving (Liu et al., 2025). By selecting the best responses to back propagate, GRPO reduces the need for extensive human annotations. However, most current tasks using GRPO to self-improve models use clear defined correctness as rewards, which limits the scope to further self-improve models on long-form generations that does not have definite clear correctness rules.

Challenges of free-form and open-ended evaluations and training. Evaluating long-form and open-ended generation remains a significant challenge in LLMs (Krishna et al., 2021; Chen et al., 2019). Unlike short-form or rule-based tasks—such as math problems or short-form question answering—that provide clear-cut correctness signals, long-form outputs like open-ended questions, summaries, or dialogues lack binary ground truths. Evaluating such outputs necessitates consideration of multiple qualitative dimensions, including coherence, factual accuracy, structure, and overall helpfulness (Chiang et al., 2024; Fabbri et al., 2021).

Traditional automatic metrics like ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), BERTScore (Zhang et al., 2020), which rely on surface-level token overlap or embedding similarity, often fail to capture the semantic, organizational and pragmatic qualities of long-form responses across different dimensions. These metrics no longer align well with human judgments, particularly when evaluating aspects like logical consistency or the depth of reasoning (Chen et al., 2019).

To address these limitations, LLM-as-a-judge is used to evaluate the outputs of other models. This approach enables more flexible and semantically informed assessments, such as pairwise comparisons or Likert-scale ratings, which often show higher correlation with human evaluations than traditional metrics (Chiang et al., 2024; Gu et al., 2025; Zheng et al., 2023). However, LLM judges introduce significant computational overhead, as GRPO requires generating multiple outputs per prompt and processing them using another LLM, thereby increasing memory usage and latency for training (Luo et al., 2025). This can be prohibitive for many users, especially those without access to substantial computational resources.

There is a line of work using annotated human preference or human rating data to finetune small language models or small size LLMs to judge and evaluate long-form responses (Kim et al., 2024; Yang et al., 2024; Chen et al., 2020). Open-resource language model judges have been used for various applications such as evaluating and ranking models (Li et al., 2024; Krumdick et al., 2025), few has explored using them as verifiable rewards in the LLM training process for long-form generations, suggesting this as an underexplored area.

7 Conclusion

RLVR especially GRPO has been a success for its ability to fully leverage LLMs' abilities to self-improve without massive amount of labeled data on many rule-based evaluation tasks. However, extending GRPO study on long-form and open-ended generation has been underexplored for the challenges of evaluating long-form responses. We propose using a small fine-tuned language model (PrefBERT) to evaluates long-form responses with different dimensions and semantic quality evaluation as reward signals for long-form generations of GRPO and show that model trained with PrefBERT generates responses with overall better quality than models trained with traditional metrics such as ROUGE and BERTScore or a generalized preference reward model, even close to the quality of larger models of the same backbone. Our work shows the potential of applying more efficient and robust verifiable reward design into the GRPO pipeline for models to self-improve its long-form and open-ended generations beyond using traditional reward metrics. Future work can expand upon current work on more diverse open-ended generation tasks such as training more efficient and stronger verifiable reward models and apply them on creative writings, creative research and design, or open-ended math problems.

8 Ethics

Our annotation does not involve in collecting annotators' private information and does not involve in extensive tool usage. Thus, our annotation is exempted by the Institutional Review Board (IRB) annotation protocol.

9 Limitations

We are the first to demonstrate that fine-tuned evaluation language models can effectively leverage the capabilities of LLMs for evaluating and improving long-form, open-ended generations. However, this work has not fully explored the potential of GRPO and reward design for enabling self-improvement in LLMs on such complex tasks. A key limitation of our study is that we did not train or use a larger and more powerful language model (e.g., 7B-scale) to serve as a verifiable reward provider, primarily due to computational constraints. Larger evaluators, while potentially offering more reliable and semantically accurate rewards, significantly increase GPU memory usage and slow down training. We hypothesize that incorporating a stronger evaluator to provide high-quality, verifiable rewards could unlock the full potential of GRPO for aligning LLMs on open-ended tasks.

References

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online. Association for Computational Linguistics.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2020. Mocha: A dataset for training and evaluating generative reading comprehension metrics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Cheng-Han Chiang and Hung yi Lee. 2023a. Can large language models be an alternative to human evaluations? *Preprint*, arXiv:2305.01937.

Cheng-Han Chiang and Hung yi Lee. 2023b. A closer look into automatic evaluation using large language models. *Preprint*, arXiv:2310.05657.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: an open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

DeepSeek-AI, Daya Guo, and et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *Preprint*, arXiv:2104.08758.

Aarohi Srivastava et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Preprint*, arXiv:2206.04615.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Preprint*, arXiv:2007.12626.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *Preprint*, arXiv:1907.09190.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

Jian Hu, Xibin Wu, Zilin Zhu, Xianyu, Weixun Wang, Dehao Zhang, and Yu Cao. 2024. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las

Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *Preprint*, arXiv:2405.01535.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. *Preprint*, arXiv:2103.06332.

Michael Krumdick, Charles Lovering, Varshini Reddy, Seth Ebner, and Chris Tanner. 2025. No free labels: Limitations of llm-as-a-judge without human grounding. *Preprint*, arXiv:2503.05061.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2023. Longform: Effective instruction tuning with reverse instructions. *Preprint*, arXiv:2304.08460.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. *Preprint*, arXiv:2403.13787.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven C. H. Hoi. 2022. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *Preprint*, arXiv:2207.01780.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *Preprint*, arXiv:1606.01541.

Zongxia Li, Ishani Mondal, Yijun Liang, Huy Nghiem, and Jordan Lee Boyd-Graber. 2024. Pedants: Cheap but effective and interpretable answer equivalence. *Preprint*, arXiv:2402.11161.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *Preprint*, arXiv:2503.20783.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. Notion Blog.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Léonard Hussenot, Orgad Keller, Nikola Momchev, Sabela Ramos, Piotr Stanczyk, Nino Vieillard, Olivier Bachem, Gal Elidan, Avinatan Hassidim, Olivier Pietquin, and Idan Szpektor. 2023. Factually consistent summarization via reinforcement learning with textual entailment feedback. *Preprint*, arXiv:2306.00186.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Kun Shao, Zhentao Tang, Yuanheng Zhu, Nannan Li, and Dongbin Zhao. 2019. A survey of deep reinforcement learning in video games. *Preprint*, arXiv:1912.10944.

Stack Exchange contributors. 2025. Stack Exchange. `https://{site}.stackexchange.com/questions/{question_id}`. [Online; accessed 5-May-2025].

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023a. Alpaca: A strong, replicable instruction-following model. `https://crfm.stanford.edu/2023/03/13/alpaca.html`. [Online; accessed 5-May-2025].

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. Stanford alpaca:

An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *Preprint*, arXiv:2412.13663.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Wikipedia contributors. 2025. Wikipedia, the free encyclopedia. [Online; accessed 5-May-2025].

Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. 2023. Pairwise proximal policy optimization: Harnessing relative feedback for llm alignment. *Preprint*, arXiv:2310.00212.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024. Regularizing hidden states enables learning generalizable reward model for llms. *Preprint*, arXiv:2406.10216.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. *Preprint*, arXiv:2305.11206.

A Appendix

Training Prompt Template

The user asks a question, and the Assistant answers it. The assistant provides the user with the answer that strictly follows the following guidelines. The answer should be enclosed within <answer> </answer> tags, respectively, i.e., <answer> ANSWER HERE </answer>. Your answer should follow these rubric criterias:

Rubric:

Factual Accuracy: The answer must be factually correct and does not contradict the reference answer.

Relevance and Completeness: The answer should directly address the specific question, covering all essential aspects.

Clarity and Organization: The answer should be well-structured, coherent, and easy to follow.

Conciseness: The answer should avoid unnecessary repetition and be as clear and succinct as possible.

Completeness: The answer is complete and not repetitive.

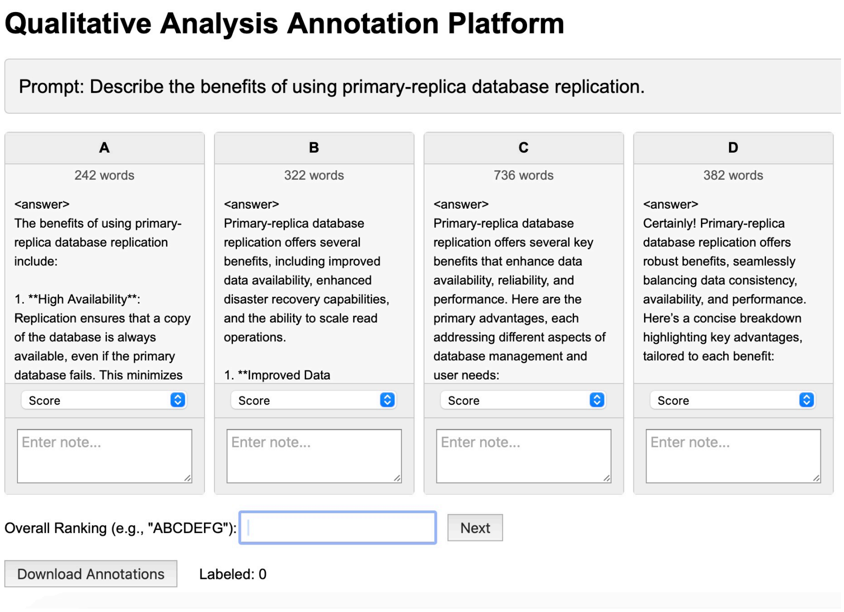
Response Format rules:- Always start your response with <answer> tag and end with </answer>.- Do not include any text or commentary before the opening <answer> tag and after the closing </answer> tag.example, your response follow this format:<answer>[Your final detailed answer goes here]</answer>

Question: {question}

Table 5: Training prompt template for LLMs to generate responses.

B Evaluation Prompt Template

We show the point-wise evaluation prompt template (Table B) and pairwise-preference prompt-template (Table 7).



Qualitative Analysis Annotation Platform

Prompt: Describe the benefits of using primary-replica database replication.

A

242 words

<answer>
The benefits of using primary-replica database replication include:

1. **High Availability**: Replication ensures that a copy of the database is always available, even if the primary database fails. This minimizes

Score

Enter note...

B

322 words

<answer>
Primary-replica database replication offers several benefits, including improved data availability, enhanced disaster recovery capabilities, and the ability to scale read operations.

1. **Improved Data

Score

Enter note...

C

736 words

<answer>
Primary-replica database replication offers several key benefits that enhance data availability, reliability, and performance. Here are the primary advantages, each addressing different aspects of database management and user needs:

Score

Enter note...

D

382 words

<answer>
Certainly! Primary-replica database replication offers robust benefits, seamlessly balancing data consistency, availability, and performance. Here's a concise breakdown highlighting key advantages, tailored to each benefit:

Score

Enter note...

Overall Ranking (e.g., "ABCDEFG"):

Next

Download Annotations

Labeled: 0

Figure 2: Our annotation tool for response quality annotation. Annotators will be displayed with the question prompt, the answers for the seven models, where they need to slide due to limited screen width. Annotators can then put their Likert scores (1-5) and comments or notes for each response, and then finally rank the responses based on their preferences and ratings.

Point-wise Evaluation Template

You will be given a user question, a reference answer, and a system answer. Your task is to provide an overall rating scoring how well the system answer addresses the user question against the reference answer. Give your answer as an integer on a scale of 1 to 5, where 1 means that the system answer is not informative, and 5 means that the answer addresses the question according to the criteria below.

Rubric:

Factual Accuracy: The answer must be factually correct and does not contradict the reference answer.

Relevance and Completeness: The answer should directly address the specific question, covering all essential aspects.

Clarity and Organization: The answer should be well-structured, coherent, and easy to follow.

Conciseness: The answer should avoid unnecessary repetition and be as clear and succinct as possible.

Completeness: The answer is complete and not repetitive.

Please base your overall rating on how well the system answer performs in these areas.

Question: {question}

Reference Answer: {reference_answer}

System Answer: {answer}

Please be as strict and as critical and harsh as possible.

Provide your feedback as follows:

Feedback:::

Final rating: (your rating, as an integer between 1 and 5)

Table 6: Prompt template for point-wise evaluation.

Pairwise Preference Evaluation Template

You are a fair judge assistant tasked with providing clear, objective feedback based on specific criteria, ensuring each assessment reflects the absolute standards set for performance.

Your task is to provide your preferred response as either A or B. please strictly follow the output format as: Feedback: Reason why you choose this answer[RESULT] A or B</s>

Rubric:

Factual Accuracy: The answer must be factually correct and does not contradict the reference answer.

Relevance and Completeness: The answer should directly address the specific question, covering all essential aspects.

Clarity and Organization: The answer should be well-structured, coherent, and easy to follow.

Conciseness: The answer should avoid unnecessary repetition and be as clear and succinct as possible.

Completeness: The answer is complete and not repetitive.

Write a detailed feedback that assess the quality of two responses strictly based on the given score rubric, not evaluating in general. After writing a feedback, choose a better response between Response A and Response B. You should refer to the score rubric.

Question: {question}

Reference Answer: {reference_answer}

Answer A: {answer_A}

Answer B: {answer_B}

Please be as strict and as critical and harsh as possible.

Provide your feedback as follows:

Feedback:::

Final rating: (your rating, as an integer between 1 and 5)

Table 7: Prompt template for pairwise evaluation.