Foundational Autoraters: Taming Large Language Models for Better Automatic Evaluation

Tu Vu[♠ ♡ *] **Kalpesh Krishna**[◇ *] **Salaheddin Alzubi**[†]
Chris Tar[♠ ‡] **Manaal Faruqui**[◇ ‡] **Yun-Hsuan Sung**[♠ ‡]
[♠]Google DeepMind, [◇]Google, [♡]Virginia Tech
{ttvu,kalpeshk}@google.com

Abstract

As large language models (LLMs) evolve, evaluating their output reliably becomes increasingly difficult due to the high cost of human evaluation. To address this, we introduce FLAMe, a family of Foundational Large Autorater Models. FLAMe is trained on a diverse set of over 100 quality assessment tasks, incorporating 5M+ human judgments curated from *publicly released human evaluations*. FLAMe outperforms models like GPT-4 and Claude-3 on various held-out tasks, and serves as a powerful starting point for fine-tuning, as shown in our reward model evaluation case study (FLAMe-RM). On Reward-Bench, FLAMe-RM-24B achieves 87.8% accuracy, surpassing GPT-4-0125 (85.9%) and GPT-4o (84.7%). Additionally, we introduce FLAMe-Opt-RM, an efficient *tail-patch fine-tuning* approach that offers competitive RewardBench performance using 25× fewer training datapoints. Our FLAMe variants outperform popular proprietary LLM-as-a-Judge models on 8 of 12 autorater benchmarks, covering 53 quality assessment tasks, including RewardBench and LLM-AggreFact. Finally, our analysis shows that FLAMe is significantly less biased than other LLM-as-a-Judge models on the CoBBLEr autorater bias benchmark.[1]

1 Introduction

The growing capabilities of large language models (LLMs) present a key challenge: *How can we reliably evaluate their long-form responses?* A promising approach is to use the models themselves as autoraters. After large-scale multitask instruction tuning, LLMs can generalize to follow new human instructions (Wei et al., 2022; Sanh et al., 2022; Longpre et al., 2023; Chung et al., 2024), making them suitable for this task. This is appealing because human evaluation, while essential, is limited by subjectivity (Krishna et al., 2023a), inconsistency among raters (Karpinska et al., 2021), and the high costs of extensive evaluations (Min et al., 2023; Vu et al., 2023; Wei et al., 2024).

Training LLM autoraters on human judgments is essential for aligning them with human preferences (Ouyang et al., 2022). However, gathering these judgments is both costly and time-consuming. Reusing human evaluations from prior research is a promising approach, yet it faces challenges such as inconsistent standards, diverse criteria, inadequate documentation, and privacy or proprietary concerns. On the other hand, training autoraters on model outputs offers consistency (Jiang et al., 2024b; Kim et al., 2024b) but risks reinforcing biases and hallucinations (Gudibande et al., 2023; Muennighoff et al., 2023) and may also breach proprietary LLM service terms.[2]

To address these limitations, we curated and standardized human evaluations from prior research to create FLAMe, a collection of 102 quality assessment tasks comprising more than 5.3M total human judgments (§3). FLAMe spans a wide variety of task types, from assessing summarization quality to evaluating how well AI assistants follow user instructions. We hypothesized that training on this large and diverse data collection would enable LLM autoraters to learn robust, generalized patterns of human judgment, minimizing the impact of noisy or low-quality human judgments.

For transparency and reproducibility, we use only *publicly available human evaluation data with permissive licenses* from previous studies (§3.2). To address challenges due to the lack of standardization and documentation, we thoroughly exam-

*Tu Vu and Kalpesh Krishna contributed equally to the project leadership, design, and implementation of the work.
†Work done while at UMass Amherst.
‡Equal contribution as senior advisors.
[1]The FLAMe collection is available at https://huggingface.co/datasets/google/flame-collection.
[2]https://openai.com/policies/terms-of-use, https://policies.google.com/terms/generative-ai

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 17086–17105
November 12-16, 2024

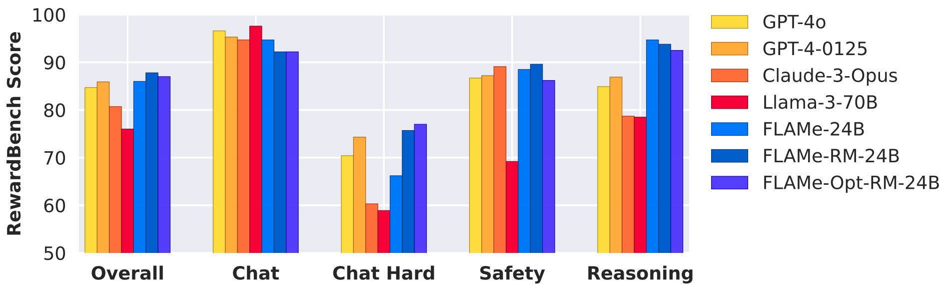

Figure 1: Our FLAMe-24B variants outperform popular proprietary LLM-as-a-Judge models like GPT-4 and Claude-3 on various autorater benchmarks, including RewardBench. As of July 15, 2024, FLAMe-RM, with an overall accuracy of 87.8%, was the top-performing generative model trained exclusively on permissively licensed data on RewardBench, surpassing GPT-4-0125 (85.9%) and GPT-4o (84.7%).

ined the associated research and consulted the original authors to clarify ambiguities or inconsistencies, spending 3-4 hours per dataset. Inspired by T5 (Raffel et al., 2020), we unify all tasks into a *text-to-text* format, with manually crafted task definitions and evaluation instructions. This simple and adaptable data format facilitates effective transfer learning, allowing our models to interpret and respond consistently to various tasks (Figure 2).

Our approach can be viewed as developing general-purpose LLM autoraters for various quality assessment tasks. We show that training an instruction-tuned LLM, PaLM-2-24B (Anil et al., 2023), on our FLAMe collection improves zero-shot generalization to a wide range of held-out tasks, outperforming models like GPT-4, Claude-3, and Llama-3 on many tasks. This demonstrates that our large-scale multitask instruction tuning enhances the model's general-purpose quality assessment capabilities.

Motivated by these results, we explore FLAMe's effectiveness as a powerful starting point for fine-tuning on targeted downstream applications, using reward model evaluation on RewardBench (Lambert et al., 2024) as a case study (FLAMe-RM). Specifically, we slightly fine-tune FLAMe on a mixture of four datasets with human pairwise preference judgments, covering chat, reasoning, and safety. The resulting FLAMe-RM-24B model achieves a notable performance boost on RewardBench, reaching an accuracy of 87.8% (up from 86.0%). As of July 15, 2024, it was *the top-performing generative model trained solely on permissively licensed data*, outperforming GPT-4-0125 (85.9%) and GPT-4o (84.7%); see Figure 1.

Additionally, we present FLAMe-Opt-RM, a computationally efficient method for optimizing our FLAMe multitask mixture for targeted reward model evaluation on RewardBench. Using a novel *tail-patch fine-tuning* technique, we evaluate the impact of each dataset on specific RewardBench distributions, enabling us to determine the optimal dataset proportions for our mixture. Fine-tuning the initial instruction-tuned PaLM-2-24B on this optimized mixture yields competitive RewardBench performance (87.0%) compared to FLAMe (86.0%), using 25× fewer training datapoints.

Overall, our FLAMe variants outperform all popular proprietary LLM-as-a-Judge models we consider on 8 out of 12 autorater evaluation benchmarks (1 held-in and 11 held-out), covering 53 quality assessment tasks, including RewardBench and LLM-AggreFact (Tang et al., 2024). Finally, our analysis shows that FLAMe variants are significantly less biased than other popular LLM-as-a-Judge autoraters on the CoBBLEr bias benchmark (Koo et al., 2023), demonstrating greater robustness to changes in pairwise ordering, response length, and irrelevant context.

In summary, our main contributions are: 1) **Data Collection**: We curated and standardized human evaluations from permissively licensed datasets, creating a collection of over 100 diverse quality assessment tasks with 5M+ human judgments. To facilitate future research, we release our data collection at `https://huggingface.co/datasets/google/flame-collection`; 2) **LLM Autoraters**: We show that our data collection can be used for training general-purpose LLM autoraters (FLAMe) and optimizing them for specific applications (FLAMe-RM and FLAMe-Opt-RM). Our models outperform popular proprietary LLM-as-a-Judge models on 8 out of 12 autorater benchmarks, covering 53 tasks, including RewardBench and LLM-AggreFact; and 3) **Computationally Efficient Multitask Training**: We propose a tail-

```
"""Input format."""

INSTRUCTIONS:
"""Task definition and evaluation instructions."""
title: Is all of the information in the summary fully attributable to the source article?
description: In this task, you will be shown a summary and a source news article on which the summary is based. Your task is to
evaluate whether the summary is attributable to the source article. Answer 'Yes' if all the information in the summary is fully
supported by the source article, or 'No' if any information in the summary is not supported by the source article. Provide an
explanation for your answer.
output_fields: answer, explanation

CONTEXT:
"""Input fields for context, each starting with a label indicating its type or purpose and is separated by a newline, for example:
'article': <article>
'summary': <summary>
"""
article: Tower Hamlets Council said it would sell Draped Seated Woman after "unprecedented" budget cuts. The work has not yet
been valued but a Moore sold for £17m earlier this year. The council said the rising threat of metal theft and vandalism made it too
expensive to insure if it was on show. The sculpture was bought by the former London County Council for £6,000 in 1960. The
bronze sculpture, nicknamed Old Flo, was installed on the Stifford council estate in 1962 but was vandalised and moved to the
Yorkshire Sculpture Park in 1997. A council spokesperson said: "With unprecedented cuts to council budgets, the council finds
itself in a difficult situation and being forced to make hard decisions."
summary: A Moore sculpture of a woman sitting on a concrete plinth is to be sold.
```

```
"""Target format."""

EVALUATION:
"""Target fields, each starting with a label indicating its type or purpose and is separated by a newline, for example:
'choice': <choice>
'explanation': <explanation>
"""
answer: No
explanation: The detail that the woman is "sitting on a concrete plinth" is not in the article.
```

Figure 2: We unify all quality assessment tasks into a *text-to-text* format, with manually crafted task definitions and evaluation instructions. Each training example consists of an input-target pair: the input provides task-specific context, while the target contains the expected human evaluation. This format can be easily adapted to novel tasks.

patch fine-tuning method that optimizes our multitask mixture for specific distributions, achieving competitive performance with significantly reduced compute.

2 Related Work

Below, we discuss existing literature in the space of autoraters, drawing connections to FLAMe.

Automatic Evaluation Metrics: Traditional metrics like BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) focus on lexical overlap between model output and human references. In the BERT era (Devlin et al., 2019), newer methods use pretrained models to measure distributional similarity (Zhao et al., 2019; Zhang et al., 2020) or token probabilities (Thompson and Post, 2020; Yuan et al., 2021). Several approaches assess divergence between text distributions (Gehrmann et al., 2019; Pillutla et al., 2021). Other work fine-tunes models on human ratings for specific tasks like machine translation (Sellam et al., 2020; Rei et al., 2020; Fernandes et al., 2023), summarization (Durmus et al., 2020; Deutsch et al., 2021; Goyal and Durrett, 2021), and QA (Chen et al., 2020; Lin et al., 2022). Unlike task-specific metrics, FLAMe is trained on diverse quality assessment tasks and can adapt to new tasks during inference.

LLM-as-a-Judge Autoraters: Prior work has used LLMs as judges to assess LLM capabilities on various benchmarks (Liu et al., 2023a; Fu et al., 2024; Bai et al., 2023; Wang et al., 2023a; Chiang et al., 2023; Chiang and Lee, 2023; Bubeck et al., 2023). However, these models tend to favor their own generated responses (Liu et al., 2023a; Panickssery et al., 2024; Liu et al., 2023b; Bai et al., 2023), showing biases toward factors like length, order, and entity preference (Koo et al., 2023). In contrast, FLAMe is trained on a broad range of human evaluations, enabling it to learn unbiased, generalized patterns of human judgment (§6.1). Additionally, FLAMe is not tasked with evaluating its own responses, avoiding self-preference bias.

Recent work has also trained general-purpose LLM autoraters. Jiang et al. (2024b) introduce TIGERScore, a Llama-2 model trained on GPT-4-generated error analysis data. Similar methods include InstructScore (Xu et al., 2023b), Prometheus (Kim et al., 2024a), and Prometheus-2 (Kim et al., 2024b). Unlike these, we rely solely on open-source human evaluations instead of model outputs. FLAMe significantly outperforms Prometheus-2 on RewardBench (see Table 2).

Appendix A has related work on reward models.

3 The FLAMe Collection

We curated 5.3M human judgments across 102 training tasks, with an additional 53 tasks reserved for evaluation (§5.1). Appendix B lists our datasets. Our data covers various task types and LLM capabilities (§3.2-3.3). We manually crafted task definitions and evaluation instructions, converting all tasks into a unified format (§3.4).

3.1 Task Definition

A "task" refers to a specific assignment where a model evaluates aspects of a text (e.g., a machine-generated summary), alongside its context (the original article), based on given criteria (Figure 2). Each task has its own definition and evaluation guidelines. Multiple tasks can be derived from a single dataset.[3] Additionally, similar tasks from different datasets are treated as separate. Based on this definition, FLAMe has 102 distinct tasks.

3.2 Principles for Data Collection

Our principles for data selection are as follows:

Public, Open-source Data: We use only permissively licensed datasets from HuggingFace (Lhoest et al., 2021), TensorFlow,[4] or the original authors' GitHub repositories.

Human Annotations: We only use human-labeled annotations, avoiding those generated by models like GPT-4 due to potential inaccuracies and legal concerns (Gudibande et al., 2023; Muennighoff et al., 2023).

Diverse Task Types: To improve model generalizability, we collect datasets from a diverse set of task types (see breakdown in Figure 3): 1) **Pairwise Evaluation**: Tasks that involve comparing two responses to determine a preference (e.g., *"Which response, A or B, is more helpful?"*); 2) **Pointwise Evaluation**: Tasks that involve evaluating specific attributes of individual responses (e.g., *"Rate the overall coherence of the response on a 5-point Likert scale."*); 3) **Classification**: Tasks that involve categorizing responses into predefined categories (e.g., *"Does the model output follow the instructions? (Yes/No)"*); and 4) **Open-ended Evaluation**: Tasks that require free-form, unrestricted answers (e.g., *"Is the summary fully attributable to the source article? Provide a brief explanation."*).

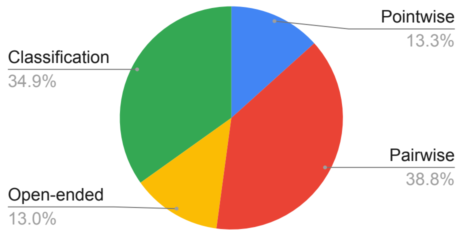

Figure 3: FLAMe data collection breakdown by task type, showing the percentage of datapoints (out of 5.3M) for each task type. Over half of FLAMe is dedicated to standard pairwise (*"Which response is better?"*) and pointwise (*"Rate the response on a Likert scale."*) evaluation. The remainder includes classification (e.g., *"Is the summary fully attributable to the source article? (Yes/No)"*) and open-ended evaluation (e.g., *"Explain why response A is better than response B."*).

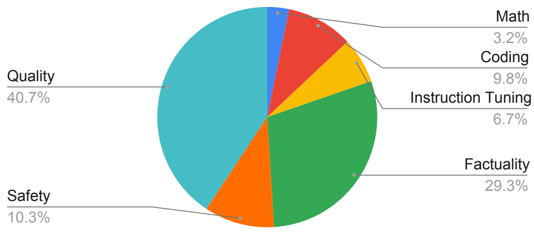

Figure 4: FLAMe data collection breakdown by LLM capability, showing the percentage of datapoints (out of 5.3M) for each LLM capability. We focus on standard LLM evaluation pillars: general response quality, factuality, safety, coding, and math. Additionally, we incorporate non-evaluation instruction tuning data (e.g., LIMA) to maintain FLAMe's general-purpose instruction-following capabilities.

Various LLM Capabilities: We select datasets from the literature that evaluate various LLM capabilities, including factuality, safety, reasoning, instruction-following, long-form generation, creativity, attribution, and coding (§3.3).

3.3 LLM Capabilities Covered by FLAMe

FLAMe encompasses key LLM capabilities, as outlined below (see breakdown in Figure 4).

General Response Quality: We assess LLM response quality using datasets that measure attributes like helpfulness, coherence, fluency, cre-

[3] For example, HelpSteer (Wang et al., 2023b) includes human annotations for attributes like helpfulness and correctness, enabling separate tasks for each attribute.

[4] https://www.tensorflow.org/datasets

ativity, complexity, and verbosity. These include: Summary Comparisons (SummFeedback) (Stiennon et al., 2020), LMSYS Chatbot Arena conversations (Zheng et al., 2023), HH RLHF Helpfulness (Bai et al., 2022a), WebGPT (Nakano et al., 2021), SummEval (Fabbri et al., 2021), News Summary Evaluation (Goyal et al., 2022), SHP (Ethayarajh et al., 2022), BeaverTails Helpfulness (Ji et al., 2023), SEAHORSE (Clark et al., 2023), HelpSteer (Wang et al., 2023b), etc. For instruction-following abilities, we use datasets such as GENIE (Khashabi et al., 2022), InstruSum (Liu et al., 2024), and riSum (Skopek et al., 2023).

Factuality/Attribution: To measure hallucinations in LLM-generated responses, we use several datasets that evaluate factual accuracy and grounding (e.g., checking if claims are supported by source documents). These include: XSum Hallucination (Maynez et al., 2020), QAGS (Wang et al., 2020), WikiBio Hallucination (Manakul et al., 2023), FRANK (Pagnoni et al., 2021), FactScore (Min et al., 2023), VitaminC (Schuster et al., 2021), HaluEval (Li et al., 2023), Q^2 (Honovich et al., 2021), FaithDial (Dziri et al., 2022a), DialFact (Gupta et al., 2022), BEGIN (Dziri et al., 2022b), and MNLI (Williams et al., 2018), etc.[5]

Mathematical Reasoning: We create data to help FLAMe distinguish between correct and incorrect solutions to mathematical problems. Using PRM800K (Lightman et al., 2024), we extract pairs of human vs. incorrect LLM-generated solutions, along with pairs of (*correct, incorrect*) LLM-generated solutions.

Coding: We train FLAMe for code evaluation. Using Code Contests (Li et al., 2022a), CommitPack (Muennighoff et al., 2023), and COFFEE (Moon et al., 2023), we create pairs of (*correct, buggy*) programs based on coding problems or GitHub issues. FLAMe learns to identify the correct program or fix across programming languages like Python, JavaScript, Java, C++, Go, and Rust.

Safety: Developing safe AI assistants for public use is crucial. To improve safety evaluation, we train FLAMe to identify harmless responses. Our training data includes tasks from HH RLHF Harmlessness (Bai et al., 2022a), HH RLHF Red Teaming (Ganguli et al., 2022), BeaverTails QA-Classification and Harmlessness (Ji et al., 2023).

[5]We reformulate natural language inference as quality assessment because it naturally aligns with attribution.

Instruction Tuning: Finally, to preserve our models' instruction-following capabilities, we incorporate instruction tuning data from human-written response datasets, including LIMA (Zhou et al., 2023), PRM800K IF (Lightman et al., 2024),[6] and TULU-2 (Ivison et al., 2023).[7]

3.4 Unified Task Format

We standardize our datasets into a unified text-to-text format. This preprocessing step takes around 3-4 hours per dataset and includes several key tasks: 1) **Comprehensive Review and Author Consultations:** We carefully review the associated research and consult with the original authors to clarify ambiguities or inconsistencies; 2) **Data Collection:** We gather all relevant data files from the corresponding HuggingFace, TensorFlow, or GitHub repositories; 3) **Data Extraction:** We extract data fields with human quality assessments; 4) **Task Definitions and Evaluation Instructions:** We write detailed task definitions and evaluation instructions for each task, ensuring consistency and standardization, while adhering to any available instructions provided to the original annotators. These instructions help FLAMe identify input/output formats and specific aspects to assess; and 5) **Text-to-Text Format Conversion:** We convert all tasks into a unified format (Figure 2). Task definitions, evaluation instructions, and desired output fields are listed under an `INSTRUCTIONS` block, while input and target field values are placed under `CONTEXT` and `EVALUATION` blocks, respectively. This format is easily adaptable to new tasks.

4 Model

We fine-tune the instruction-tuned PaLM-2-24B on the FLAMe collection to create general-purpose LLM autoraters that can be prompted to perform various tasks. We train three FLAMe variants: 1) **FLAMe**—trained with examples-proportional mixture weights (Raffel et al., 2020); 2) **FLAMe-RM**—initialized with FLAMe and fine-tuned on a balanced mixture of four pairwise evaluation datasets covering chat, reasoning, and safety (§4.2); and 3) **FLAMe-Opt-RM**—trained with RewardBench-optimized mixture weights (§4.3).

[6]We train FLAMe to produce the ground truth solutions.

[7]We only use TULU-2 instruction tuning subsets with human-written responses, including FLAN, CoT, Open Assistant 1, Science literature, and Hardcoded (see Section 2 in Ivison et al., 2023 for details).

4.1 General-purpose Autoraters (FLAMe)

Our baseline FLAMe model is trained using supervised multitask training on the instruction-tuned PaLM-2-24B for 30K steps. We use examples-proportional mixture weights, capping each task at a maximum of 2^{16} examples to avoid oversampling large datasets. FLAMe shows significant generalization improvements across various held-out tasks, outperforming models like GPT-4, Claude-3, and Llama-3 on many tasks (see Figure 1 and Table 1). This supports our hypothesis that large-scale multitask instruction tuning enhances general-purpose quality assessment capabilities.

4.2 FLAMe for Reward Model Evaluation (FLAMe-RM)

We delve deeper into FLAMe's potential as a powerful starting point for fine-tuning on specific downstream applications, focusing on reward model evaluation as a case study. We create FLAMe-RM by fine-tuning FLAMe on a balanced mixture of four pairwise evaluation datasets: HelpSteer (Wang et al., 2023b), PRM800K (Lightman et al., 2024), CommitPack (Muennighoff et al., 2023), and HH-RLHF Harmlessness (Bai et al., 2022a). Since FLAMe is already trained on these datasets, we fine-tune for only 50 steps. FLAMe-RM significantly boosts FLAMe's RewardBench accuracy from 86.0% to 87.8%. As of July 15, 2024, FLAMe-RM-24B became the top-performing generative model trained solely on permissively licensed data, surpassing both GPT-4-0125 (85.9%) and GPT-4o (84.7%); see Figure 1 and Table 1.

4.3 Optimizing FLAMe for RewardBench (FLAME-Opt-RM)

Our baseline approach requires extensive training to attain strong performance on certain downstream tasks like RewardBench (Figure 5). This may stem from suboptimal mixture weights that undersample beneficial tasks. To address this, we introduce a tail-patch ablation strategy that evaluates each dataset's impact on targeted distributions, allowing efficient adjustment of all mixing weight hyperparameters. Fine-tuning the instruction-tuned PaLM-2-24B on this optimized mixture for just 5000 steps achieves competitive RewardBench performance (87.0%) compared to the baseline FLAMe (86.0%), using 25× fewer training datapoints.

We optimized our multitask mixture directly based on RewardBench performance due to the absence of a development set. Our early experiments showed weak correlations between RewardBench and other held-out tasks, making it hard to create a reliable *proxy* development set. Our goal here is not to achieve state-of-the-art RewardBench results but to demonstrate how to optimize our multitask mixture for specific distributions.[8] Furthermore, FLAMe-Opt-RM's strong performance across other held-out tasks (Table 1) indicates that it was not overfitted to RewardBench.

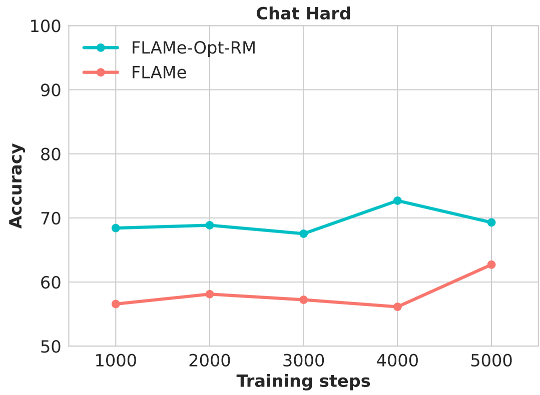

Figure 5: Comparison of FLAMe-Opt-RM and FLAMe during the first 5000 training steps based on RewardBench Chat Hard performance. FLAMe-Opt-RM, with optimized mixture weights, reaches significantly higher Chat Hard scores faster than FLAMe. For reference, FLAMe scores 66.2 at 30K steps. See Figure 6 in Appendix C for RewardBench safety results.

Tail-patch Ablations: Assigning the right mixing weight for each task in our multitask mixture is challenging due to the large number of tasks. Instead, we assess each task's impact on targeted distributions and use that to assign weights. First, we select a checkpoint that has been partially trained on our vanilla mixture, showing decent but not optimal RewardBench performance.[9] Then, we perform a brief fine-tuning stage ("tail-patch") on each individual training task, limited to 3000 training steps. This is a one-time process for each downstream application and can be done with smaller models to reduce computational costs.

A Re-weighted Mixture: After training a tail-patch on each task, we rate its impact on each RewardBench category using four ratings: *Helpful* (+2, significant and stable improvement), *Some-*

[8] Longer training or additional fine-tuning (as with FLAMe-RM) further improved performance, though we did not submit these results to the official leaderboard.

[9] We hypothesize that using a partially trained checkpoint, rather than the initial one, is better for tail-patch ablations, since the model has already been exposed to multitask data and is familiar with its overall distribution.

what helpful (+1, slight improvement), *No clear effect* (0, minimal change), *Harmful* (-1, significant drop). We then group tasks into seven bundles: *Generally helpful* (tasks with a total rating of ≥ 5), *Category-specific*, one for each of the five RewardBench categories (most beneficial tasks for each category with performance exceeding a threshold τ),[10] and *Others* for the remaining tasks.

We assign fixed mixing weights to each bundle: $w_{general}$=100K for *Generally helpful*, $w_{specific}$=30K for each *Category-specific* bundle, and w_{others}=3K for *Others*. If a task belongs to multiple bundles, its final weight is the sum of the mixture weights from each bundle.[11] An exception to this rule is that we prioritize the top two tasks in three underperforming categories——Chat Hard, Coding, and Safety——each assigned a fixed weight of $w_{top\_specific}$=200K. These values were initially set based on our intuition and not extensively tuned.

4.4 Training Details

We initialize both FLAMe and FLAMe-Opt-RM with PaLM-2-24B (Anil et al., 2023), instruction-tuned on the Flan collection (Longpre et al., 2023), and train for 30K and 5K steps, respectively. FLAMe is further fine-tuned for 50 steps to create FLAMe-RM. Our models are trained using T5X (Roberts et al., 2023) with the Adam optimizer (Kingma and Ba, 2015), a learning rate of 0.0001, and a dropout rate of 0.05. FLAMe is trained on 256 Cloud TPU chips with a batch size of 32, whereas FLAMe-RM and FLAMe-Opt-RM use 128 Cloud TPU chips with a batch size of 8.[12]

5 Experiments

We compare FLAMe to several popular LLM-as-a-Judge autoraters (§5.2) using a suite of 12 autorater benchmarks (1 held-in and 11 held-out), covering a total of 53 quality assessment tasks (§5.1). Overall, FLAMe variants outperform all LLM-as-a-Judge autoraters on 8 out of 12 benchmarks (§5.3).

5.1 Evaluation Datasets

We use a variety of held-in and held-out tasks. Each task is cast into our unified task format (§3.4). For benchmarks with multiple categories (e.g., RewardBench, LLM-AggreFact), we use the same prompt instructions across categories. To minimize API costs, we randomly sample 256 examples per evaluation task,[13] except for RewardBench, where results are reported for the full set.

5.1.1 Held-in Evaluations

HelpSteer (Wang et al., 2023b): We assess FLAMe's performance in rating helpfulness, correctness, coherence, complexity, and verbosity, using HelpSteer's validation data.

5.1.2 Held-out Evaluations

RewardBench (Lambert et al., 2024): RewardBench is a popular benchmark for evaluating reward models via pairwise preference tasks, where models select the better response between two options based on a given prompt. It incorporates 23 datasets, covering four categories—Chat, Chat Hard, Reasoning (Math + Coding), and Safety.[14]

LLM-AggreFact (Tang et al., 2024): This benchmark integrates ten attribution datasets to assess the grounding capabilities of autoraters. The autorater evaluates whether a claim is fully supported by a given document.

Other Benchmarks: In addition to RewardBench and LLM-AggreFact, we include a diverse set of held-out pointwise and pairwise evaluation benchmarks, including Summary Comparisons (SummFeedback) (Stiennon et al., 2020);[15] Helpful, Honest, and Harmless Alignment (HHH) (Askell et al., 2021); AlpacaFarm (Dubois et al., 2023); Paraphrase Evaluation (Dipper) (Krishna et al., 2023b); Sequence Continuation Preference (RankGen) (Krishna et al., 2022); Poem Preference (CoPoet) (Chakrabarty et al., 2022); Literary Translation Comparisons (LitTrans) (Karpinska and Iyyer, 2023); Long-form QA Evaluation (LFQAEval) (Xu et al., 2023a); and Text Continuation Preference (ContrSearch) (Su and Xu, 2022).

5.2 Evaluated Models

We compare our models to the original instruction-tuned PaLM-2-24B, which was not trained on

[10] We separate Math and Coding for the Reasoning category, and use thresholds of $\tau = 95\%, 66\%, 99.8\%, 84\%, 85\%$ for Chat, Chat Hard, Math, Coding, and Safety, respectively.

[11] For example, if a task is generally helpful and specifically beneficial for both Chat Hard and Safety, it contributes $w_t = w_{general} + 2 \times w_{specific}$ to the final mixture.

[12] cloud.google.com/tpu/docs/v5e-training, https://cloud.google.com/tpu/docs/v3

[13] For tasks with fewer than 256 examples, we use the full evaluation set.

[14] We excluded the "Prior sets" of RewardBench because three out of the four datasets were used in training FLAMe.

[15] During training, we used only pairwise ratings from the dataset and reserved pointwise ratings for evaluation.

Model	Reward Bench	LLM AggreFact	Summ Feedback	Alpaca Farm	Rank Gen	Co Poet	Contr Search	HHH	Dipper	Lit Trans	LFQA Eval	Help Steer
Llama-3-70B-Instruct	76.1	76.1	50.8	53.9	65.6	53.6	53.1	91.9	42.8	60.5	71.1	39.7
Mixtral-8×7B	77.8	73.8	43.8	55.1	63.3	52.9	56.6	90.0	42.2	61.7	71.5	34.0
GPT-3.5-turbo-0125	64.5	70.0	15.6	55.5	58.2	49.0	57.5	85.5	45.0	54.3	69.9	32.0
Claude-3-Opus	80.7	79.2	31.6	49.6	55.1	49.0	45.1	**94.6**	**50.6**	71.1	71.1	41.3
GPT-4-0125	85.9	80.6	46.5	49.6	62.5	56.9	55.8	**94.6**	45.0	67.6	**77.0**	37.9
GPT-4o	84.7	80.2	30.9	50.4	66.0	55.6	57.5	92.3	45.6	**72.7**	75.0	40.1
our models												
PaLM-2-24B	62.9	54.8	13.3	52.3	58.2	54.2	46.0	85.5	48.3	62.5	70.3	20.0
FLAMe-24B	86.0	**81.1**	48.0	**58.2**	62.1	53.6	**69.9**	91.4	48.3	67.2	74.2	**48.4**
FLAMe-RM-24B	**87.8**	80.8	**53.1**	57.8	65.2	**57.5**	57.5	91.0	47.8	67.6	72.7	46.6
FLAMe-Opt-RM-24B	87.0	80.2	52.3	53.1	**69.5**	52.9	48.7	89.1	48.3	69.5	69.5	35.9

Table 1: Performance of FLAMe compared to popular LLM-as-a-Judge autoraters across various autorater benchmarks. Overall, FLAMe variants outperform all LLM-as-a-Judge autoraters on 8 out of 12 benchmarks, including RewardBench and LLM-AggreFact. See §5.1 for the sources of our benchmarks.

FLAMe, to isolate the effects of instruction tuning and FLAMe training. We also evaluate several popular LLM-as-a-Judge autoraters, including Llama-3-70B-Instruct (Meta, 2024), Mixtral 8×7B (Jiang et al., 2024a), Claude-3-Opus (Anthropic, 2024), GPT-3.5-turbo-0125 (OpenAI, 2024a), GPT-4-0125 (OpenAI, 2024b), and GPT-4o (OpenAI, 2024c).[16] Additionally, we include several models from the official RewardBench leaderboard, notably Gemini-1.5-Pro (Reid et al., 2024), Prometheus-2-8×7B (Kim et al., 2024b), ArmoRM-Llama-3-8B-v0.1 (Wang et al., 2024a), and NVIDIA's Nemotron-4-340B-Reward and Llama-3-70B-SteerLM-RM (Wang et al., 2024b).

5.3 Main Results

Table 1 shows our main results across all evaluation benchmarks. RewardBench and LLM-AggreFact results are shown in Table 2 and Table 6, respectively. Below, we first provide an overview of these results before analyzing them in more detail:

FLAMe Variants Outperform all LLM-as-a-Judge Autoraters on 8 out of 12 Benchmarks: Table 1 shows FLAMe's strong generalization to various held-out tasks, highlighting its effectiveness as a versatile LLM autorater. FLAMe provides significant gains over the initial instruction-tuned PaLM-2-24B. Remarkably, our models outperform all state-of-the-art LLM-as-a-Judge autoraters on 8 out of 12 benchmarks. FLAMe variants outperform the next-best model by significant margins on several held-out benchmarks, including ContrSearch (69.9 vs. 57.5 for GPT-4o/GPT-3.5-turbo-0125),

Model	Avg.	Chat	Chat Hard	Safety	Reason
custom classifiers on the official RewardBench leaderboard					
ArmoRM-Llama-3	90.4	**96.9**	76.8	90.5	97.3
Nemotron-340B	**92.2**	95.8	**87.1**	92.2	93.6
Cohere May 2024	89.5	96.4	71.3	92.7	**97.7**
Llama-3-SteerLM	89.0	91.3	80.3	**93.7**	90.6
generative models on the official RewardBench leaderboard					
GPT-3.5-turbo	64.5	92.2	44.5	62.3	59.1
Prometheus-8x7B	75.3	93.0	47.1	83.5	77.4
Llama-3-70B-Inst	76.0	**97.6**	58.9	69.2	78.5
Mixtral-8×7B	77.8	95.0	64.0	73.4	78.7
Claude-3-Opus	80.7	94.7	60.3	89.1	78.7
Gemini-1.5-Flash	82.1	92.2	63.5	87.7	85.1
GPT-4o	84.7	96.6	70.4	86.7	84.9
GPT-4-0125	85.9	95.3	74.3	87.2	86.9
Gemini-1.5-Pro	**88.1**	92.3	**80.6**	87.5	92.0
our generative autorater models					
PaLM-2-24B	62.9	89.9	61.2	55.3	45.2
FLAMe-24B	86.0	94.7	66.2	88.5	**94.7**
FLAMe-RM-24B	**87.8**	92.2	75.7	**89.6**	93.8
FLAMe-Opt-24B	87.0	92.2	77.0	86.2	92.5

Table 2: As of July 15, 2024, FLAMe-RM-24B outperforms other generative models on the RewardBench leaderboard, achieving the best score (87.8%) among models trained solely on permissively licensed data.

RankGen (69.5 vs. 66.0 for GPT-4o), AlpacaFarm (58.2 vs. 55.5 for GPT-3.5-turbo-0125), SummFeedback (53.1 vs. 50.8 for Llama-3-70B-Instruct), and RewardBench (87.8 vs. 85.9 for GPT-4-0125). Additionally, our models achieve the best held-in performance on HelpSteer (48.4 vs. 41.3 for Claude-3-Opus).

On the other hand, FLAMe variants lag behind proprietary models on several benchmarks, including HHH (91.4 vs. 94.6 for GPT-4-0125/Claude-3-Opus), LitTrans (69.5 vs. 72.7 for GPT-4o), and LFQAEva (74.2 vs. 77.0 for GPT-4-0125), indicating that these models may have been optimized for these capabilities.

[16] For fair comparison, we use the same FLAMe prompt instructions when evaluating LLM-as-a-Judge baselines. For better reproducibility, we set the temperature to 0 and generate up to 1024 tokens across all models.

Autorater	Avg. (↓)	Order (↓)	Compassion (↓)	Length (↓)	Egocentric (↓)	Bandwagon (↓)	Attention (↓)
Random	0.30	0.50	0.50	0.00	0.25	0.25	0.25
baselines reported in Koo et al. (2023)							
Falcon-40B	0.31	0.77	0.27	0.09	**0.05**	0.28	0.40
Cohere-54B	0.41	0.50	0.65	0.10	0.27	0.82	0.14
Llama-2-70B	0.19	0.61	0.26	0.12	0.06	0.04	0.03
InstructGPT	0.45	0.38	0.48	0.16	0.28	0.85	0.54
ChatGPT	0.45	0.41	0.66	0.13	0.58	0.86	0.06
GPT-4	0.31	0.23	0.79	0.06	0.78	**0.00**	**0.00**
our models							
FLAMe-24B	**0.13**	**0.08**	0.09	0.03	0.38	0.18	**0.00**
FLAMe-RM-24B	**0.13**	0.11	**0.08**	0.02	0.40	0.17	**0.00**
FLAMe-Opt-RM-24B	0.15	0.15	0.14	**0.00**	0.41	0.17	**0.00**

Table 3: Autorater bias analysis on the CoBBLEr bias benchmark from Koo et al. (2023). **Lower values indicate better or less biased autoraters** across all columns. Overall, FLAMe variants exhibit significantly less bias compared to popular LLM-as-a-Judge autoraters like GPT-4. Compared to Table 2 in Koo et al. (2023), we combine first/last numbers for Order/Compassion, report $|\text{bias} - 0.5|$ for Length, and only report the order setup in Egocentric.

FLAMe Variants are among the Most Powerful Generative Models on RewardBench: Our results in Table 2 show that FLAMe variants rank among the top generative models on the official RewardBench leaderboard,[17] demonstrating strong performance in all categories: Chat, Chat Hard, Safety, and Reasoning. Notably, FLAMe-RM-24B achieves an overall score of 87.8%, the highest among generative models trained solely on permissively licensed data, surpassing GPT-4-0125 (85.9) and GPT-4o (84.7). As of July 15, 2024, FLAMe-RM-24B ranked second among generative models (below Gemini-1.5-Pro) and sixth overall. We provide an analysis of length and token biases found in RewardBench in Appendix E. Additionally, we discuss our LLMAggreFact results in Appendix D.

6 Further Analysis of FLAMe

In this section, we depart from the typical focus on analyzing the effect of factors like model size, data size, and data quality in multitask learning, which have been extensively studied (Raffel et al., 2020; Longpre et al., 2023). Instead, we examine potential biases in our LLM autoraters. We find that our models are significantly less biased than popular LLM-as-a-Judge autoraters. In Appendix F, we further demonstrate FLAMe's potential utility for AI development, particularly in identifying high-quality responses for code generation.

6.1 Autorater Bias Analysis

A common criticism of LLM-as-a-Judge autoraters is their bias towards certain judgments (Liu et al., 2023a; Panickssery et al., 2024; Liu et al., 2023b; Bai et al., 2023). Here, we evaluate FLAMe variants on the CoBBLEr autorater bias benchmark (Koo et al., 2023).

CoBBLEr measures six types of biases in LLM autoraters: 1) **Order:** Does the autorater favor a particular response position? 2) **Compassion:** Does the autorater's judgment change when using the LLM's actual name, like "GPT-4" instead of aliases like "Model A"? 3) **Length:** Does the autorater prefer longer or shorter outputs? 4) **Egocentric:** Does the autorater favor outputs it generated itself? 5) **Bandwagon:** Is the autorater influenced by statements like *"90% of people prefer response A"*? 6) **Attention:** Does irrelevant context, such as *"Response A is about cats."* distract the autorater? We reformat the original (*prompt,response*) pairs from Koo et al. (2023) into our unified FLAMe format (Figure 2) and compare FLAMe variants to other LLM-as-a-Judge autoraters, including GPT-4, reported in Koo et al. (2023).

Table 3 shows that FLAMe variants exhibit significantly lower bias compared to GPT-4 and other autoraters, with an average bias of 0.13 vs. 0.31 for GPT-4 (lower is better). FLAMe matches or outperforms GPT-4 across all six bias categories. These results demonstrate FLAMe's effectiveness as a robust and reliable autorater.

7 Conclusion

We curated and standardized human evaluations from permissively licensed datasets, compiling a data collection of over 100 quality assessment tasks with 5M+ human judgments. We demonstrate that this collection can be used for training general-purpose LLM autoraters and optimizing them for specific downstream applications. Our models outperform popular proprietary LLM autoraters on 8 out of 12 autorater benchmarks, covering 53 tasks.

[17] https://huggingface.co/spaces/allenai/reward-bench

Limitations and Future work

Evaluating LLMs is challenging due to evolving evaluation standards and the need to assess new LLM capabilities. Expanding our data collection with open-source contributions could address this issue. Additionally, our models, trained primarily on English data with a context length of 2048 tokens, might not perform well on multilingual (Freitag et al., 2021) or long-context (Kim et al., 2024c; Karpinska et al., 2024) quality assessment tasks. Finally, in this work, we train our models in a supervised multitask fashion. Exploring alternative training approaches such as RLHF and DPO is a promising direction for future work.

Ethical Considerations and Risks

All considerations and risks outlined by prior work for pretrained and instruction-tuned LLMs (Chowdhery et al., 2022; Anil et al., 2023) apply to LLM autoraters. We recommend following standard practice for responsible development of these models (Achiam et al., 2023; Gemini et al., 2023; Reid et al., 2024). Additionally, LLM autoraters raise new risks due to increased quality assessment capabilities. First, our models can inherit and amplify biases from human evaluations, leading to unfair or discriminatory outcomes. For instance, the model may replicate biases related to race, gender, or other sensitive attributes from the training data, potentially harming certain groups. Second, overreliance on LLM autoraters risks automating decisions that need human understanding and empathy. To mitigate these risks, transparency in model development and use, along with robust measures like bias audits, data anonymization, and incorporating diverse perspectives, is essential for promoting fairness, accountability, and trustworthiness.

Acknowledgments

We are grateful to Jie Ren, Denny Zhou, and Tania Bedrax-Weiss for their comments on this manuscript. We thank Mohit Iyyer, Daniel Cer, Elizabeth Clark, Jeremiah Liu, Balaji Lakshminarayanan, Clara Huiyi Hu, Aliaksei Severyn, Adam Sadovsky, Yonghui Wu, Quoc Le, Slav Petrov, Séb Arnold, Taylan Bilal, Noah Constant, Colin Raffel, Nan Hua, Marzena Karpinska, Yixiao Song, Tuhin Chakrabarty, the Gemini model quality team, the Descartes team at Google, and the UMass NLP group for useful discussions and valuable feedback at different stages of this project. We thank the authors of the datasets used in this work, especially Niklas Muennighoff, Hyungjoo Chae, Mounica Maddela, Tanya Goyal, and Yuanhao Wu, for their helpful suggestions and for answering our questions. Finally, we thank Grady Simon, Chung-Ching Chang, Sho Kannan, Gustavo Hernandez Abrego, and the T5X team for their assistance with the codebase, implementation, and computational resources.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

AI Anthropic. 2024. Introducing the next generation of claude.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. 2023. Benchmarking foundation models with language-model-as-an-examiner. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, volume 36, pages 78142–78167.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, volume 31. Curran Associates, Inc.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Tuhin Chakrabarty, Vishakh Padmakumar, and He He. 2022. Help me write a poem - instruction tuning as a vehicle for collaborative poetry writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6848–6863.

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)*, pages 6521–6532.

Bei Chen, Fengji Zhang, Anh Nguyen, Daoguang Zan, Zeqi Lin, Jian-Guang Lou, and Weizhu Chen. 2023. Codet: Code generation with generated tests. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 15607–15631.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research (JMLR)*, 25(70):1–53.

Elizabeth Clark, Shruti Rijhwani, Sebastian Gehrmann, Joshua Maynez, Roee Aharoni, Vitaly Nikolaev, Thibault Sellam, Aditya Siddhant, Dipanjan Das, and Ankur Parikh. 2023. SEAHORSE: A multilingual, multifaceted dataset for summarization evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9397–9413.

Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2021. Towards question-answering as an automatic metric for evaluating the content quality of a summary. *Transactions of the Association for Computational Linguistics (TACL)*, 9:774–789.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 4171–4186.

Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi. 2022a. Is GPT-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7250–7274.

Yao Dou, Chao Jiang, and Wei Xu. 2022b. Improving large-scale paraphrase acquisition and generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9301–9323.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, volume 36, pages 30039–30069.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5055–5070.

Nouha Dziri, Ehsan Kamalloo, Sivan Milton, Osmar Zaiane, Mo Yu, Edoardo M. Ponti, and Siva Reddy. 2022a. FaithDial: A faithful benchmark for information-seeking dialogue. *Transactions of the Association for Computational Linguistics (TACL)*, 10:1473–1490.

Nouha Dziri, Hannah Rashkin, Tal Linzen, and David Reitter. 2022b. Evaluating attribution in dialogue systems: The BEGIN benchmark. *Transactions of the Association for Computational Linguistics (TACL)*, 10:1066–1083.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research (PMLR)*, pages 5988–6008.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics (TACL)*, 9:391–409.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André Martins, Graham Neubig, Ankush Garg, Jonathan Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation. In *Proceedings of the Eighth Conference on Machine Translation (WMT)*, pages 1066–1083.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics (TACL)*, 9:1460–1474.

Daniel Fried, Armen Aghajanyan, Jessy Lin, Sida Wang, Eric Wallace, Freda Shi, Ruiqi Zhong, Scott Yih, Luke Zettlemoyer, and Mike Lewis. 2023. Incoder: A generative model for code infilling and synthesis. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 6556–6576.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL)*, pages 111–116.

Team Gemini, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 1449–1462.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*.

Prakhar Gupta, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. DialFact: A benchmark for fact-checking in dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3785–3801.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7856–7870.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First Quora Dataset release: Question pairs.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, volume 36, pages 24678–24704.

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, et al. 2024a. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Dongfu Jiang, Yishan Li, Ge Zhang, Wenhao Huang, Bill Yuchen Lin, and Wenhu Chen. 2024b. TIGERScore: Towards building explainable metric for all text generation tasks. *Transactions on Machine Learning Research (TMLR)*.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using Mechanical Turk to evaluate open-ended text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1265–1285.

Marzena Karpinska and Mohit Iyyer. 2023. Large language models effectively leverage document-level context for literary translation, but critical errors persist. In *Proceedings of the Eighth Conference on Machine Translation (WMT)*, pages 419–451.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. One thousand and one pairs: A" novel" challenge for long-context language models. *arXiv preprint arXiv:2406.16264*.

Daniel Khashabi, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A. Smith, and Daniel Weld. 2022. GENIE: Toward reproducible and standardized human evaluation for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 11444–11458.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024a. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations (ICLR)*.

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

Yekyung Kim, Yapei Chang, Marzena Karpinska, Aparna Garimella, Varun Manjunatha, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024c. Fables: Evaluating faithfulness and content selection in book-length summarization. *arXiv preprint arXiv:2404.01261*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2023. Benchmarking cognitive biases in large language models as evaluators. *arXiv preprint arXiv:2309.17012*.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research (PMLR)*, pages 17506–17533.

Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023a. LongEval: Guidelines for human evaluation of faithfulness in long-form summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. RankGen: Improving text generation with large ranking models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 199–232.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 4940–4957.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023b. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, volume 36, pages 27469–27500.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pages 175–184.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6449–6464.

Ruosen Li, Teerth Patel, and Xinya Du. 2024. PRD: Peer rank and discussion improve large language model based evaluations. *Transactions on Machine Learning Research (TMLR)*.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. 2022a. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097.

Zichao Li, Prakhar Sharma, Xing Han Lu, Jackie Cheung, and Siva Reddy. 2022b. Using interactive feedback to improve the accuracy and explainability of question answering systems post-deployment. In *Findings of the Association for Computational Linguistics: ACL 2022 (ACL Findings)*, pages 926–937.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Workshop of Text Summarization Branches Out (WS)*, pages 74–81.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3214–3252.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2511–2522.

Yiqi Liu, Nafise Sadat Moosavi, and Chenghua Lin. 2023b. Llms as narcissistic evaluators: When ego inflates evaluation scores. *arXiv preprint arXiv:2311.09766*.

Yixin Liu, Alexander Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. 2024. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. In *Findings of the Association for Computational Linguistics: NAACL 2024 (NAACL Findings)*, pages 4481–4501.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research (PMLR)*, pages 22631–22648.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2023. LENS: A learnable evaluation metric for text simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 16383–16408.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9004–9017.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1906–1919.

AI Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 12076–12100.

Seungjun Moon, Yongho Song, Hyungjoo Chae, Dongjin Kang, Taeyoon Kwon, Kai Tzu-iunn Ong, Seung-won Hwang, and Jinyoung Yeo. 2023. Coffee: Boost your code llms by fixing bugs with feedback. *arXiv preprint arXiv:2311.07215*.

Niklas Muennighoff, Qian Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro Von Werra, and Shayne Longpre. 2023. Octopack: Instruction tuning code large language models. *arXiv preprint arXiv:2308.07124*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2023. Codegen: An open large language model for code with multi-turn program synthesis. In *The Eleventh International Conference on Learning Representations (ICLR)*.

OpenAI. 2024a. GPT-3.5 Turbo.

OpenAI. 2024b. GPT-4 Turbo and GPT-4.

OpenAI. 2024c. Hello GPT-4o.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John

Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 27730–27744.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 4812–4829.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076*.

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 of the Association for Computational Linguistics (ACL)*, pages 311–318.

Zarana Parekh, Jason Baldridge, Daniel Cer, Austin Waters, and Yinfei Yang. 2021. Crisscrossed captions: Extended intramodal and intermodal semantic similarity judgments for MS-COCO. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (EACL)*, pages 2855–2870.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, volume 34, pages 4816–4828.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 36.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR 2020)*, 21(140):1–67.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Adam Roberts, Hyung Won Chung, Gaurav Mishra, Anselm Levskaya, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, Curtis Hawthorne, Aitor Lewkowycz, Alex Salcianu, Marc van Zee, Jacob Austin, Sebastian Goodman, Livio Baldini Soares, Haitang Hu, Sasha Tsvyashchenko, Aakanksha Chowdhery, Jasmijn Bastings, Jannis Bulian, Xavier Garcia, Jianmo Ni, Andrew Chen, Kathleen Kenealy, Kehang Han, Michelle Casbon, Jonathan H. Clark, Stephan Lee, Dan Garrette, James Lee-Thorp, Colin Raffel, Noam Shazeer, Marvin Ritter, Maarten Bosma, Alexandre Passos, Jeremy Maitin-Shepard, Noah Fiedel, Mark Omernick, Brennan Saeta, Ryan Sepassi, Alexander Spiridonov, Joshua Newlan, and Andrea Gesmundo. 2023. Scaling up models and data with t5x and seqio. *Journal of Machine Learning Research (JMLR)*, 24(377):1–8.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations (ICLR)*.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 624–643.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7881–7892.

Ondrej Skopek, Rahul Aralikatte, Sian Gooding, and Victor Carbune. 2023. Towards better evaluation of instruction-following: A case-study in summarization. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 221–237.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, volume 33, pages 3008–3021.

Yixuan Su and Jialu Xu. 2022. An empirical study on contrastive search and contrastive decoding for open-ended text generation. *arXiv preprint arXiv:2211.10797*.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. *arXiv preprint arXiv:2404.10774*.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, et al. 2023. Freshllms: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5008–5020.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024a. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023a. Is ChatGPT a good NLG evaluator? a preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop (NewSum)*, pages 1–11.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024b. Helpsteer2: Open-source dataset for training top-performing reward models. *arXiv preprint arXiv:2406.08673*.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, et al. 2023b. Helpsteer: Multi-attribute helpfulness dataset for steerlm. *arXiv preprint arXiv:2311.09528*.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics (TACL)*, 7:625–641.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations (ICLR)*.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, et al. 2024. Long-form factuality in large language models. *arXiv preprint arXiv:2403.18802*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (NAACL)*, pages 1112–1122.

Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Cheng Niu, Randy Zhong, Juntong Song, and Tong Zhang. 2023a. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. *arXiv preprint arXiv:2401.00396*.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023b. Fine-grained human feedback gives better rewards for language model training. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, volume 36, pages 59008–59033.

Fangyuan Xu, Junyi Jessy Li, and Eunsol Choi. 2022. How do we answer complex questions: Discourse structure of long-form answers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3556–3572.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023a. A critical evaluation of evaluations for long-form question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3225–3245.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. 2023b. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5967–5994.

Xinyan Yu, Sewon Min, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. CREPE: Open-domain question answering with false presuppositions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 10457–10480.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, volume 34, pages 27263–27277.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations (ICLR)*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (ACL)*, pages 1298–1308.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, volume 36, pages 46595–46623.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan 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. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, volume 36, pages 55006–55021. Curran Associates, Inc.

Appendix

A Related Work on Reward Models

Our work relates to the development of reward models (RMs) used to align LLMs with human preferences using reinforcement learning with human feedback (RLHF) (Ouyang et al., 2022; Korbak et al., 2023). In RLHF, human preference data is either used to train standalone discriminative RMs, or directly fed into LLMs via algorithms like DPO (Rafailov et al., 2024) or SLiC-HF (Zhao et al., 2023). While we evaluate our models as RMs in our RewardBench experiments (§5), there are key distinctions: (1) RMs primarily rely on pairwise preference data,[18] while our models use diverse task types in a unified format; (2) RMs optimize for overall preference, whereas our models can be prompted to judge specific aspects of responses (e.g., safety).

B List of Training Datasets in FLAMe

Table 5 shows the list of datasets used in our study.

C Additional Results for FLAME-Opt-RM

See Figure 6 for RewardBench safety results.

D Performance of FLAMe on LLM-Aggrefact

Table 6 presents a breakdown of our attribution results on LLM-AggreFact (Tang et al., 2024), categorized into four common use cases: 1) LLM-FactVerify: fact verification of LLM-generated responses, 2) Wiki-FactVerify: evaluating correctness of Wikipedia claims, 3) Summarization: assessing faithfulness of summaries, and 4) Long-form QA: evaluating long-form answers to questions. FLAMe variants outperform all other models in three out of the four categories (LLM-FactVerify, Wiki-FactVerify, and Summarization). FLAMe-24B achieves the highest overall performance of 81.1, while the next-best baseline model GPT-4-0125 obtains a score of 80.6. In long-form QA attribution evaluation, our best model FLAMe-Opt-RM underperforms compared to GPT-4-0125 (74.8 vs. 77.3), aligning with our findings in Table 1.

[18] A notable exception is RL*AI*F (Bai et al., 2022b), which asks the model to critique its responses based on a constitution.

E Analyzing Length and Token Bias in RewardBench

In this section, we provide an analysis of length (Appendix E.1) and token (Appendix E.2) bias issues identified in the RewardBench benchmark. Given these issues, we encourage future work to evaluate LLM autoraters on a wide variety of benchmarks (such as our evaluation suite in §5), rather than relying solely on RewardBench.

E.1 Length Bias in RewardBench

Table 4 highlights length bias in RewardBench. Overall, RewardBench shows significant imbalance across categories regarding length: Chat Hard, Math, and Coding favor shorter outputs, while Chat leans towards longer outputs. An adversarial submission might strategically select longer or shorter outputs based on prompt categories to achieve higher scores, without necessarily reflecting a genuinely strong preference model.

RewardBench Category	% Preference for Longer Outputs
Chat	79.1%
Chat Hard	29.6%
Math	6.5%
Coding	35.7%
Safety	41.9%

Table 4: Summary of length bias in RewardBench. Overall, we find that four out of five RewardBench categories show a strong preference towards either longer or shorter outputs.

E.2 Token Bias in RewardBench

Besides length bias, we identified token bias in the Math and Safety categories of RewardBench. In Safety, favored responses significantly leaned towards phrases like *"I'm sorry"*, which suggest hedged responses. The word "sorry" appeared nearly 23% more frequently in preferred responses compared to non-preferred ones. Similarly, the Math split exhibited token bias, where tokens such as "i", "can", "need", "to", "find" were predominantly found in rejected responses.

F Using FLAMe to Re-rank Decoded Outputs

In this section, we explore the application of our LLM autoraters in selecting optimal outputs from multiple responses, a method known as *"Best-of-N"* sampling (Nakano et al., 2021; Krishna et al., 2022). Using FLAMe for re-ranking, we assess its impact on code generation performance with the HumanEval Python programming benchmark (Chen et al., 2021). We conduct experiments by re-ranking 10 code samples generated by three models: OpenAI's davinci-002, InCoder-6B (Fried et al., 2023), and CodeGen-16B (Nijkamp et al., 2023) using a round-robin competition, and then measuring performance with the top-ranked code sample.[19] Results in Table 7 show that FLAMe provides significant gains in pass@1 accuracy across all three models. Notably, FLAMe improves CodeGen-16B's pass@1 from 21.2 to 31.1, closing nearly 40% of the gap to the Oracle ranker (46.9).

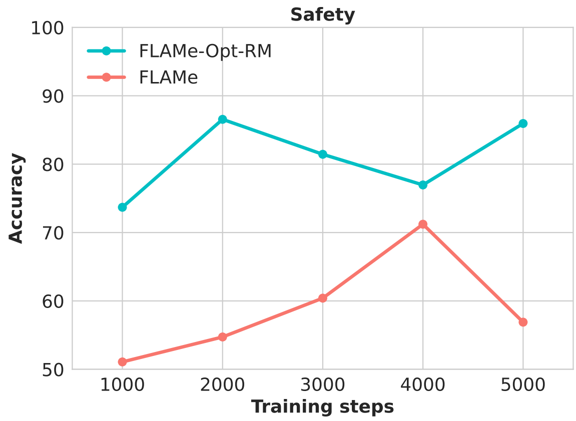

Figure 6: Comparison of FLAMe-Opt-RM and FLAMe during the first 5000 training steps based on RewardBench Safety performance. FLAMe-Opt-RM, with optimized mixture weights, reaches significantly higher Safety scores faster than FLAMe. For reference, FLAMe scores 88.5 at 30K steps.

[19] We use relatively weak LLMs from Chen et al. (2023) for two main reasons: (1) to assess the potential benefits of re-ranking with FLAMe, and (2) HumanEval has been extensively used to develop newer LLMs.

Capability	Dataset	Source	Output Format
General Response Quality	BeaverTails Helpfulness	Ji et al. (2023)	Pairwise
	HH RLHF Helpfulness	Bai et al. (2022a)	Pairwise
	Hurdles LFQA	Krishna et al. (2021)	Pairwise
	LMSYS Chatbot Arena conversations	Zheng et al. (2023)	Pairwise
	MAUVE	Pillutla et al. (2021)	Pairwise
	News Summary Evaluation	Goyal et al. (2022)	Pairwise
	PRD	Li et al. (2024)	Pairwise
	SHP	Ethayarajh et al. (2022)	Pairwise
	HelpSteer	Wang et al. (2023b)	Pairwise, Pointwise
	Summary Comparisons	Stiennon et al. (2020)	Pairwise, Pointwise
	GENIE	Khashabi et al. (2022)	Pairwise, Pointwise, Generative
	Fine-grained RLHF	Wu et al. (2023b)	Pairwise, Classification
	InstruSum	Liu et al. (2024)	Pairwise, Classification
	WebGPT	Nakano et al. (2021)	Pairwise, Generative
	LENS	Maddela et al. (2023)	Pointwise
	SummEval	Fabbri et al. (2021)	Pointwise
	riSum	Skopek et al. (2023)	Pointwise, Classification
	FeedbackQA	Li et al. (2022b)	Pointwise, Generative
	CoLA	Warstadt et al. (2019)	Classification
	SEAHORSE	Clark et al. (2023)	Classification
	CREPE	Yu et al. (2023)	Classification, Generative
	Scarecrow	Dou et al. (2022a)	Classification, Generative
	Validity LFQA	Xu et al. (2022)	Classification, Generative
Factuality/Attribution	MOCHA	Chen et al. (2020)	Pointwise
	Sentence Similarity - C$\times$C	Parekh et al. (2021)	Pointwise
	Sentence Similarity - STS-B	Cer et al. (2017)	Pointwise
	WikiBio Hallucination	Manakul et al. (2023)	Pointwise
	BEGIN	Dziri et al. (2022b)	Classification
	DialFact	Gupta et al. (2022)	Classification
	FActScore	Min et al. (2023)	Classification
	FRANK	Pagnoni et al. (2021)	Classification
	FaithDial	Dziri et al. (2022a)	Classification
	HaluEval	Li et al. (2023)	Classification
	MNLI	Williams et al. (2018)	Classification
	MultiPIT	Dou et al. (2022b)	Classification
	PAWS	Zhang et al. (2019)	Classification
	Q^2	Honovich et al. (2021)	Classification
	QAGS	Wang et al. (2020)	Classification
	QQP	Iyer et al. (2017)	Classification
	VitaminC	Schuster et al. (2021)	Classification
	RAGTruth	Wu et al. (2023a)	Classification
	ESNLI	Camburu et al. (2018)	Classification, Generative
	XSum Hallucination	Maynez et al. (2020)	Generative
Mathematical Reasoning	PRM800K	Lightman et al. (2024)	Pairwise
Coding	Code Contests	Li et al. (2022a)	Pairwise
	COFFEE	Moon et al. (2023)	Pairwise
	CommitPack	Muennighoff et al. (2023)	Pairwise
	CommitPack - Bugs	Muennighoff et al. (2023)	Pairwise
Safety	BeaverTails Harmlessness	Ji et al. (2023)	Pairwise
	HH RLHF Harmlessness	Bai et al. (2022a)	Pairwise
	HH RLHF Red Teaming	Bai et al. (2022a)	Pointwise
	BeaverTails QA-Classification	Ji et al. (2023)	Classification
Instruction Tuning	LIMA	Zhou et al. (2023)	Generative
	PRM800K IF	Lightman et al. (2024)	Generative
	TULU-2	Ivison et al. (2023)	Generative

Table 5: A complete list of training datasets in our FLAMe collection, including their output formats and categorized capabilities. We derive multiple tasks from certain datasets. For example, HelpSteer (Wang et al., 2023b) includes human annotations for different attributes of model responses such as Helpfulness, Correctness, Coherence, Complexity, and Verbosity, allowing us to create distinct tasks, each focused on a specific attribute.

Model	Overall	LLM-FactVerify	Wiki-FactVerify	Summarization	Long-form QA
GPT-3.5-turbo-0125	70.0	80.1	71.1	64.6	65.4
Mixtral-8×7B	73.8	73.8	50.8	78.1	76.6
Llama-3-70B-Instruct	76.1	75.3	58.4	80.3	**77.7**
Claude-3-Opus	79.2	78.6	70.6	83.8	75.0
GPT-4o	80.2	79.6	71.6	85.0	76.0
GPT-4-0125	80.6	79.6	71.6	85.3	77.3
our models					
PaLM-2-24B	54.8	34.4	28.9	68.2	71.7
FLAMe-24B	**81.1**	82.3	77.7	85.3	72.7
FLAMe-RM-24B	80.8	**82.6**	77.2	**85.4**	70.9
FLAMe-Opt-RM-24B	80.2	77.6	**81.2**	84.7	74.8

Table 6: LLM-AggreFact performance across four common use cases: LLM-FactVerify (ClaimVerify + FactCheck + Reveal), Wiki-FactVerify (WiCE), Summarization (AggreFact + TofuEval), and Long-form QA (ExpertQA + LFQA). FLAMe variants outperform all tested LLM-as-a-Judge models in three out of the four use cases. FLAMe-24B achieves the highest overall performance of 81.1, while the next-best model GPT-4-0125 scores 80.6.

Ranker	CodeGen-16B	davinci002	InCoder-6B
10 code samples re-ranked in round-robin fashion			
None	21.2	17.6	14.6
FLAMe-24B	**31.1**	22.6	**22.0**
FLAMe-RM-24B	29.9	**23.2**	21.3
FLAME-Opt-RM-24B	29.3	18.3	16.5
Oracle	46.9	63.4	29.3

Table 7: Pass@1 performance on the HumanEval coding benchmark (Chen et al., 2021). Re-ranking code samples with FLAMe variants significantly improves performance across models.