Foundational Autoraters:
Taming Large Language Models for Better Automatic Evaluation

Anonymous ACL submission

Abstract

As LLMs advance, evaluating generated text reliably becomes more challenging due to the high costs of human evaluation. To make progress toward better LLM autoraters, we introduce **FLAME**, a family of **F**oundational **L**arge **A**utorater **M**od**E**ls. FLAME is trained on our large and diverse collection of nearly 100 quality assessment tasks comprising 5M+ human judgments, curated and standardized using *publicly released human evaluations* from previous research. FLAME significantly improves generalization to a wide variety of held-out tasks, outperforming proprietary LLMs like GPT-4 and CLAUDE on many tasks. Additionally, we show that our FLAME multitask mixture can be further optimized for specific downstream applications, e.g., reward modeling evaluation, through a novel *tail-patch fine-tuning* technique. Notably, on REWARDBENCH, our model (86.7) is the top-performing generative model trained solely on permissively licensed data, outperforming both GPT-4-0125 (85.9) and GPT-4O (84.7). Our analysis reveals that FLAME is significantly less biased than popular LLM-AS-A-JUDGE models on the COBBLER cognitive bias benchmark, while effectively identifying high-quality responses for code generation. We release our FLAME data collection at this http URL.

1 Introduction

The increasing power and versatility of large language models (LLMs) bring with them a growing challenge: *How can we reliably assess their long-form outputs?* Recent research suggests a promising solution: these models themselves, after undergoing large-scale multitask instruction tuning, can generalize to follow new human instructions (Mishra et al., 2022; Wei et al., 2022; Sanh et al., 2022; Chung et al., 2024), making them suitable for use as autoraters of model outputs. This is particularly appealing because human evaluation, though crucial for assessing model performance, is limited by subjectivity (Krishna et al., 2023a), variability 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).

To align LLM autoraters with human preferences, training on human judgments is crucial (Ouyang et al., 2022). However, obtaining these judgments is costly and time-consuming. Collecting existing human evaluations from previous research seems promising but faces challenges like lack of standardization, diverse evaluation criteria, inadequate documentation, and data privacy or proprietary concerns. Using model outputs for autorater training offers consistency (Jiang et al., 2023; Kim et al., 2024) but comes with risks, such as reinforcing biases and hallucinations (Gudibande et al., 2023; Muennighoff et al., 2024). Additionally, it may violate terms of use for proprietary LLM services, which prohibit using their models' outputs to develop competing models.[1]

To address these limitations, we curated and standardized human evaluations from prior research to create FLAME, a collection of approximately 100 quality assessment tasks comprising 5M+ total human judgments (§3). FLAME spans a wide variety of task types, from assessing machine translation 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.

To ensure transparency and reproducibility, we use only *publicly available human evaluation data with permissive licenses* from previous studies (§3.1). To overcome challenges in collecting such data, which rarely adhere to a particular standard

[1] https://openai.com/policies/terms-of-use, https://policies.google.com/terms/generative-ai

and often lack documentation, we thoroughly examined the associated research (§3.3) and consulted with the original authors to address ambiguities or inconsistencies (spending 3+ hours per dataset).

We train LLM autoraters using supervised, multitask fine-tuning on our data collection. All tasks are formulated into a unified text-to-text format with manually crafted task definitions and evaluation instructions. We format examples as input-target pairs, where the input includes task-specific context and the target contains human evaluations (Figure 1). This approach facilitates effective transfer learning across tasks, allowing our models to interpret and respond uniformly. Additionally, our task format is simple, intuitive, and easily accommodates new tasks.

We demonstrate that training an instruction-tuned LLM, i.e., PALM-2 24B (Anil et al., 2023), on our FLAME collection significantly improves its performance on various quality assessment tasks, outperforming models such as GPT-4, CLAUDE, and LLAMA-3 on many held-out tasks. Additionally, we show that our FLAME multitask mixture can be further optimized for specific downstream applications, using reward modeling evaluation as a case study. Specifically, we employ a novel *tail-patch fine-tuning* technique to analyze how each dataset impacts performance on targeted distributions, i.e., REWARDBENCH (Lambert et al., 2024), allowing us to determine the optimal proportions of individual datasets in our multitask training mixture. Notably, our targeted variant FLAME-RM achieves an average accuracy of 86.7 on REWARDBENCH, surpassing both GPT-4-0125 (85.9) and GPT-4o (84.7), and achieving the highest performance among generative models trained on permissively licensed datasets. Overall, our models outperform popular LLM-AS-A-JUDGE models on 6 out of 12 autorater evaluation benchmarks, covering 53 tasks (§4.3).

Motivated by these results, we further explore whether biases exist in our autoraters, a common criticism of LLM-AS-A-JUDGE autoraters (§5.1), and their potential utility for AI development, particularly in identifying high-quality model responses (§5.2). Our analysis reveals that our models are significantly less biased than popular LLM-AS-A-JUDGE models on the COBBLER cognitive bias benchmark (Koo et al., 2023), while effectively identifying high-quality responses for code generation.

In summary, our main contributions are: (1) A curated collection of approximately 100 diverse quality assessment tasks with 5M+ human judgments, available at this http URL; (2) Our LLM autoraters, which outperform all proprietary LLM-AS-A-JUDGE models like GPT-4 and CLAUDE on 6 out of 12 benchmarks, including REWARDBENCH and LLM-AGGREFACT; and (3) A novel tail-patch fine-tuning strategy for optimizing task mixtures to specific objectives.

Our work demonstrates the potential of accessible AI solutions, which we hope will spur more fundamental research into reusable human evaluations and the development of effective and efficient LLM autoraters.

2 Related work

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

Task-specific autoraters: In the pre-LLM era, several works relied on token embedding similarities (Zhang et al., 2020) or log probabilities (Yuan et al., 2021) from pretrained models like BERT (Devlin et al., 2019) for automatic text evaluation. Other work fine-tuned models on human ratings to create autoraters for specific tasks, including machine translation (Sellam et al., 2020; Thompson and Post, 2020; Rei et al., 2020; Fernandes et al., 2023; Qin et al., 2023), text summarization (Gao et al., 2019; Durmus et al., 2020; Deutsch et al., 2021), and question answering (Chen et al., 2020; Lin et al., 2022). FLAME, unlike these task-specific autoraters, is trained on various quality assessment tasks and can be prompted at inference time to perform new tasks.

LLM-AS-A-JUDGE autoraters: With the advent of instruction tuned LLMs like GPT-4, recent work has used these models as judges (Fu et al., 2023; Gong and Mao, 2023; Bai et al., 2023) to evaluate LLM capabilities on various benchmarks, including ALPACAEVAL (Li et al., 2023c; Dubois et al., 2024), MT-BENCH (Zheng et al., 2023a), and WILDBENCH (Lin et al., 2024). However, LLM-AS-A-JUDGE autoraters tend to favor their own generated responses (Panickssery et al., 2024; Liu et al., 2023a; Bai et al., 2023), exhibiting "cognitive" biases toward aspects like length, order, and entity preference (Koo et al., 2023). In contrast, our models are trained on a large, diverse collection of human evaluations, allowing them to learn unbiased, generalized patterns of human judgment (§5.1). Unlike LLM-AS-A-JUDGE autoraters, our models are not tasked with evaluating their own responses, preventing self-preference bias.

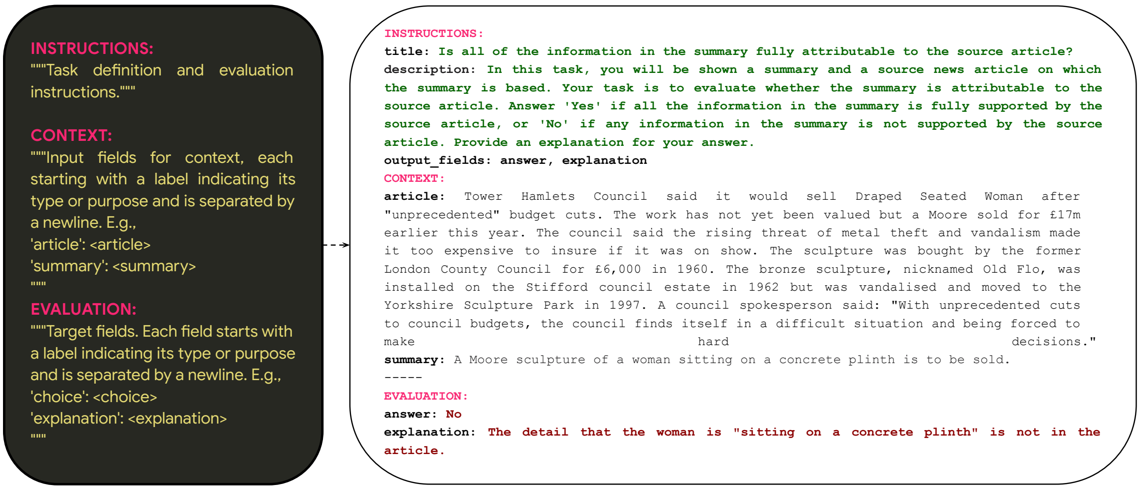

Figure 1: All of our quality assessment tasks are formulated into a unified text-to-text format with manually crafted task definitions and evaluation instructions. We format examples as input-target pairs, where the input includes task-specific context and the target contains human evaluations.

General-purpose LLM autoraters: Recent work has explored training general-purpose LLM autoraters. Jiang et al. (2023) introduced TIGERScore, a LLAMA-2 model trained on GPT-4 generated error analysis data across various tasks, including summarization and long-form QA. Similar approaches include PROMETHEUS (Kim et al., 2023), INSTRUCTSCORE (Xu et al., 2023b), and PROMETHEUS-2 (Kim et al., 2024). Unlike these efforts, our approach relies solely on open-source human evaluations instead of model outputs. We show that FLAME significantly outperforms PROMETHEUS-2 on REWARDBENCH (Table 2).

Reward models: Our work relates to reward models (RMs) used for aligning LLMs to human preferences via reinforcement learning with human feedback (RLHF, Ouyang et al., 2022; Korbak et al., 2023). In RLHF, human preference data is either used to train stand-alone discriminative RMs, or directly fed into LLMs via algorithms like DPO (Rafailov et al., 2023) or SLiC-HF (Zhao et al., 2023). While we evaluate our models as RMs in our REWARDBENCH experiments (§4), there are key distinctions: (1) RMs primarily train on pairwise preference data, whereas our models utilize diverse task types in a unified format; (2) RMs optimize for overall preference, while our models can be prompted to judge specific aspects (e.g., safety).

3 The FLAME collection

At a high level, we fine-tune instruction-tuned LLMs on our multitask mixture of standardized human evaluations. This large and diverse data collection is carefully selected to cover a wide range of LLM capabilities (§3.1-3.2). We manually crafted task definitions and evaluation instructions, reformulating all tasks into a unified text-to-text format (§3.3). We train two LLM autorater variants: one with example-proportional mixture weights (FLAME), and the other with reward modeling optimized mixture weights (FLAME-RM), determined using a tail-patch fine-tuning strategy (§3.4).

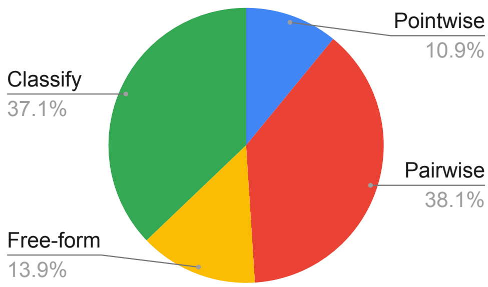

Figure 2: A breakdown of our FLAME collection by task type, with each slice representing the % of datapoints (out of 5M+) for that specific task type.

3.1 Principles for training data selection

We adhere to the following principles while choosing our datasets:

Public, open-source datasets: To ensure reproducibility, we use only permissively licensed datasets available on HUGGING FACE (Lhoest et al., 2021) or the original authors' GITHUB repositories.

Human-labeled annotations: We exclusively use datasets with human-labeled annotations, avoiding those generated by models like GPT-4 due to potential inaccuracies and legal concerns raised in recent research (Gudibande et al., 2023; Muennighoff et al., 2024).

Various task types: We gather datasets across various task types to train FLAME to generalize to new quality assessment tasks. These include pointwise evaluations (e.g., *"Rate coherence on a Likert scale of 1-5."*), pairwise evaluations (e.g., *"Which response is better, A or B?"*), classification tasks (e.g., *"Is the claim supported by the document?"*), and free-form explanation tasks (e.g., *"Which response is better? Explain your judgment."*). See Figure 2 for a breakdown.

Various LLM capabilities: We choose datasets from literature that assess diverse capabilities in modern LLMs, such as factuality, instruction following, long-form generation quality, math, coding, safety, etc. See §3.2.

3.2 Capabilities covered by FLAME mixture

Following the principles outlined in §3.1, we curated a large collection of 5M+ datapoints, composed of 97[2] training tasks (see our list of datasets in Appendix A.1). Our data collection assesses key modern LLM capabilities, detailed below (see breakdown in Figure 3):

General response quality: To evaluate LLM response quality, we use a variety of datasets that measure helpfulness, coherence, creativity, and fluency. These include pairwise comparison datasets like STANFORD SHP (Ethayarajh et al., 2022) and LMSYS (Zheng et al., 2023b), and pointwise rating datasets such as SUMMAEVAL (Fabbri et al., 2021). Additionally, to measure LLM instruction-following capabilities, we include datasets like GENIE (Khashabi et al., 2021), INSTRUSUM (Liu et al., 2023b), and RISUM (Skopek et al., 2023).

Attribution / Factuality: To address the increasing importance of measuring hallucinations in generated LLM responses, we incorporate several datasets that assess attribution or grounding, measuring whether claims or responses are supported by source documents. These include summarization evaluation (Pagnoni et al., 2021), LLM response hallucination (Li et al., 2023a), fact verification (Schuster et al., 2021), dialog faithfulness (Dziri et al., 2022a), and natural language inference (NLI) (Williams et al., 2018).[3]

Mathematical reasoning: We construct datasets to help FLAME differentiate between correct and incorrect solutions to mathematical problems. We leverage PRM800K (Lightman et al., 2024) and extract human vs incorrect LLM-generated solutions, as well as pairs of (correct, incorrect) LLM-generated solutions.

[2] An additional 53 tasks were kept for evaluation, see §4.1.

[3] We include NLI since its setup naturally fits attribution.

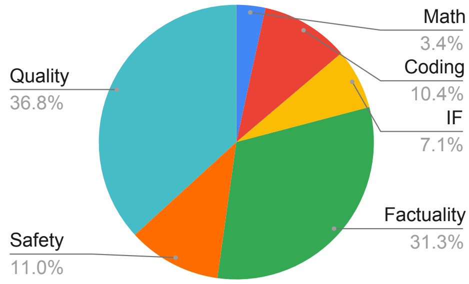

Figure 3: A breakdown of our FLAME collection by capability, with each slice representing the % of datapoints (out of 5M+) for that specific capability.

Coding: In addition to natural language evaluation, we also train FLAME to perform code evaluation. We utilize COMMITPACK (Muennighoff et al., 2024), CODE CONTESTS (Li et al., 2022a), and COFFEE (Moon et al., 2023) to construct pairs of (correct, buggy) programs in popular programming languages in response to a coding prompt or GITHUB issue. The model is trained to select the correct program in each pair.

Safety: Developing safe and harmless AI assistants for broad public use is increasingly important. To facilitate safety evaluation, we train FLAME to identify unsafe LLM responses. Our training data includes both pairwise and classification tasks from sources like BEAVERTAILS (Ji et al., 2023) and HELPFUL HARMLESS RLHF (Bai et al., 2022).

3.3 Unified FLAME prompt format

Having carefully selected our training datasets (§3.1-3.2), we then convert them into a unified text-to-text format. This involves preprocessing each dataset, which usually requires about 3-4 hours of manual work per dataset. First, we gather all relevant data files from the associated HUGGING FACE or GITHUB repository. Then, we pinpoint and extract the specific data columns containing the quality assessments conducted by human annotators. Next, we meticulously craft detailed task definitions and evaluation instructions for each quality assessment task, ensuring consistency and standardization. We leverage any available instructions provided to the original human annotators to maintain alignment with their evaluation criteria. These instructions guide the model in identifying the input and output format, as well as understanding the specific aspects it should assess. Finally, all

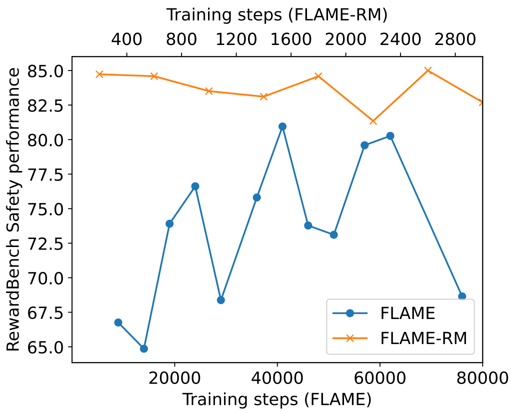

Figure 4: FLAME-RM significantly outperforms FLAME in REWARDBENCH safety performance, using 20x less compute, with improved stability, and a 2.5% performance gain.

tasks are formulated as text-to-text tasks (Figure 1). Task definitions and evaluation instructions and the list of desired output fields are placed under the `INSTRUCTIONS` block, while the input field values are placed under `CONTEXT`. This flexible text-to-text format can be easily adapted to various quality assessment tasks.

3.4 Optimizing FLAME for reward modeling evaluation (FLAME-RM)

While our vanilla FLAME mixture is effective across many tasks (§4.3), it struggles with specialized tasks like reward modeling evaluation, showing suboptimal and unstable performance across checkpoints. We attribute this instability to suboptimal mixture weights that undersample useful tasks. To address this, we introduce a novel tail-patch ablation strategy, enabling us to efficiently optimize nearly 100 hyperparameters. Using REWARDBENCH as a case study, our reward-modeling optimized mixture (FLAME-RM) achieves a 2.1% performance increase with $20\times$ less compute and significantly improved stability across checkpoints (Figure 4).

Vanilla mixture weights ("FLAME"): Our vanilla FLAME mixture assigns weights based on the number of examples per task, capped at a maximum of 2^{16} to avoid oversampling large tasks. However, as shown in Figure 4, this approach results in unstable performance for REWARDBENCH.

Tail-patch ablations to determine task usefulness: Setting the right proportion of each individual task in our mixture is non-trivial due to the nearly 100 hyperparameters. Instead, we examine the usefulness of each individual training task, and use this information for weight assignment. First, we select a checkpoint partially[4] trained on our vanilla mixture which has fair (but not optimal) performance across REWARDBENCH categories. Next, we fine-tune it *exclusively* on an individual task for only 3K steps ("tail patch"). We posit that training on a useful task would bridge the gap between fair and optimal performance.

A re-weighted mixture based on tail-patch ablations ("FLAME-RM"): After training a tail-patch on each task, we rate how helpful each training task is to each category of REWARDBENCH using one of four ratings: *Helpful* (+2, performance significantly improves and remains stable), *Somewhat helpful* (+1, performance slightly improves), *No clear effect* (0, performance is nearly unchanged), *Harmful* (-1, performance is significantly worse). We then organize tasks into seven bundles: *Generally helpful* (tasks with total highest ratings $>= 5$), *Category-specific*, one for each of the five REWARDBENCH categories (most beneficial tasks for a specific category where performance crosses a threshold τ),[5] *Others* with a fixed mixing weights for each bundles: $w_{general} = 100K$, $w_{specific} = 30K$, $w_{others} = 3K$, respectively.[6] The final weight of each task equals the total of mixing weights from the groups it belongs to. For instance, if a task is *generally helpful* and is *helpful* for CHATHARD and SAFETY, then it contributes $w_t = w_{general} + 2 * w_{specific}$ to our mixture. Since SAFETY is the most unstable category (Figure 4), we set $w_t = 250K$ for the top 3 SAFETY tasks. FLAME-RM is built on top of the initial instruction-tuned checkpoint and fine-tuned with the re-weighted mixture for only 3K steps.

3.5 Training Details

We initialize our model with the PALM-2 24B model (Anil et al., 2023), instruction tuned on the FLAN collection (Chung et al., 2024; Longpre et al., 2023). We optimize our FLAME for a total of 60K steps, while our FLAME-RM requires just 3K steps to achieve strong performance. All our models are trained using the T5X library (Roberts et al., 2023), with a learning rate of 0.0001 using the Adam optimizer (Kingma and Ba, 2015), batch size of 8, and

[4] We hypothesize that starting from a partially trained checkpoint rather than the initial checkpoint is better for tail-patch ablations, since the model has already seen some multi-task data and is familiar with its general distribution.

[5] $\tau = 95\%, 66\%, 84\%, 99.8\%, 85\%$ for CHAT, CHATHARD, CODE, MATH, and SAFETY, respectively.

[6] We note that we did not tune these weight values, all numbers were set once based primarily on our intuition.

dropout rate on 0.05. We use an input length of 2048 tokens, and target length of 1024 tokens. All models are trained on 128 CLOUD TPU V5E chips.[7]

4 Experimental Results

Having discussed the procedure used to build FLAME and FLAME-RM in §3, we now present our main experiments. We compare FLAME to several popular baseline LLM-as-judge models (§4.2) on an evaluation suite composed of 12 benchmarks and 53 tasks (§4.1). Overall, we find that FLAME outperforms proprietary LLMs like GPT-4, CLAUDE on several tasks (§4.3), despite being trained only on permissively licensed publicly available data.

4.1 Evaluation Datasets

We evaluate FLAME on a total of 12 benchmarks (composed of 53 tasks) to measure its performance as a pairwise and pointwise autorater:

RewardBench (Lambert et al., 2024) is a popular leaderboard for evaluating reward models used for RLHF. REWARDBENCH contains a suite of pairwise preference tasks, where reward models need to choose the better response among two responses to a prompt. REWARDBENCH is composed of four categories spanning many desired capabilities in LLMs (Chat, Chat-Hard, Reasoning - Math + Coding, Safety), and is built using **23** individual datasets.[8]

LLM-AggreFact (Tang et al., 2024) is a benchmark measure the attribution / grounding capabilities of autoraters. Given a reference document and a claim, the AutoRater must determine whether the claim is fully supported in the reference document. Tang et al. (2024) combine **ten** attribution datasets from recent works in LLM factuality, building a holistic benchmark for attribution evaluation.

Other pairwise evaluation tasks: Besides RewardBench, we use several pairwise preference tasks to evaluate FLAME. **None of these datasets were used while training FLAME**, so these tasks represent a true held-out setting. Our preference tasks consist of: (1) AlpacaFarm (Dubois et al., 2023); (2) RankGen (Krishna et al., 2022); (3) Contrastive Search (Su and Xu, 2022); (4) Machine Translation in literary settings, or LitMT (Karpinska and Iyyer, 2023); (5) Helpful, Honest and Harmless Alignment, or HHH-Align (Askell et al., 2021); (6) CoPoem (Chakrabarty et al., 2022); (7) Expert-LFQA (Xu et al., 2023a).

Other pointwise evaluation tasks: Additionally, we evaluate FLAME on several tasks needing Likert-scale evaluations. These include pointwise human ratings from: (1) HelpSteer (Wang et al., 2023);[9] (2) Dipper paraphrase pointwise quality evaluation (Krishna et al., 2023b); (3) Pointwise Summarization Feedback (Stiennon et al., 2020).[10]

4.2 Evaluated Models

As baselines we evaluate several popular LLM-as-a-judge models from prior work, including LLAMA-3-70B-INSTRUCT (Meta, 2024), MIXTRAL 8X7B (Jiang et al., 2024), CLAUDE 3 OPUS (Anthropic, 2024), GPT-3.5-TURBO-0125 (OpenAI, 2024a), GPT-4-0125, and OpenAI's current flagship model GPT-4O (OpenAI, 2024b).[11] We also compare against a few additional models reported on the REWARD-BENCH leaderboard (Lambert et al., 2024), notably GEMINI-1.5 (Reid et al., 2024), and PROMETHEUS-2-8X7B (Kim et al., 2024). Among **our models**, we evaluate PALM-2 24B models finetuned on FLAME and FLAME-RM (§3.4-3.5). To disentangle the effect of pretraining and FLAME training, we evaluate our initialization checkpoint (PALM-2-24B) from §3.5 which has not seen FLAME data.

4.3 Main Results

We present results across all tasks in Table 1, and REWARDBENCH in Table 2. Overall, we find that:

FLAME variants outperform all baseline on 6 out of 12 benchmarks. In Table 1 we find that despite being trained only on public data, FLAME shows strong performance in a wide variety of pairwise and pointwise tasks. Notably, it outperforms all proprietary state-of-the-art LLMs on six out of twelve tasks. This includes the LLM-Aggfact benchmark (81.2 vs next best 80.6, GPT-4-0125), confirming its utility as a cheap and effective attribution evaluator. However, FLAME notably lags behind GPT-4-0125 on ExpertQA (73.4 vs 77.0). We hypothesize this is due to the lack of expert technical knowledge in the much smaller FLAME's pa-

[7] cloud.google.com/tpu/docs/v5e-training

[8] We exclude the "Prior sets" of REWARDBENCH in our evaluation, since we used 3 of the 4 datasets in training FLAME.

[9] We leverage the five tasks in the validation split during this evaluation, and use the train split in our FLAME mixture.

[10] We leveraged only the pairwise ratings from this dataset during training, and left pointwise for evaluation.

[11] For comparable experiments with FLAME, we use the same unified prompt instructions (§3.3) while evaluating each LLM-as-a-judge baseline model. We use the default decoding hyperparameters for each API suggested by the API provider.

Model	LLM Aggfact	Reward Bench	Pointwise Tasks			Pairwise Tasks						
			H-Steer	Dipper	SumFB	Alpaca	RankG	ContS	LitMT	HHH	Copoem	EQA
LLAMA-3-70b-Inst	76.1	76.0	39.7	42.8	**50.8**	53.9	65.6	53.1	60.5	91.9	53.6	71.1
Mixtral-8x7b	73.8	77.8	34.0	42.2	43.8	55.1	63.3	56.6	61.7	90.0	52.9	71.5
GPT-3.5-turbo	70.0	64.5	32.0	45.0	15.6	55.5	58.2	57.5	54.3	85.5	49.0	69.9
Claude Opus	79.2	80.7	41.3	**50.6**	31.6	49.6	55.1	45.1	71.1	**94.6**	49.0	71.1
GPT-4-0125	80.6	85.9	40.8	45.0	46.5	49.6	62.5	55.8	67.6	**94.6**	**56.9**	**77.0**
GPT-4o	80.2	84.7	40.1	45.6	30.9	50.4	**66.3**	57.5	**72.7**	92.3	55.6	75.0
(our 24b models)												
PaLM-2-24b	54.8	62.9	20.0	48.3	13.3	52.3	58.2	46.0	62.5	85.5	54.2	70.3
FLAME-24b	80.4	84.6	**52.2**	42.8	42.2	**56.3**	65.6	**58.4**	64.1	88.2	54.2	68.4
FLAME-RM-24b	**81.2**	**86.7**	24.2	**50.6**	50.4	54.7	61.7	48.7	69.9	90.0	53.6	73.4

Table 1: Performance of FLAME compared to LLM-AS-A-JUDGE baselines on a wide variety of quality assessment tasks. Overall, we find that FLAME outperforms all proprietary LLM-AS-A-JUDGE baselines in 6 out of 12 benchmarks, including LLM-AGGREFACT and REWARDBENCH. See §4.1 for the source of each evaluation dataset.

Model	Avg	Chat	Hard	Safe	Reason
(generative baselines on RewardBench leaderboard)					
GPT-3.5-turbo	64.5	92.2	44.5	62.3	59.1
Prometheus-2	75.3	93.0	47.1	83.5	77.4
Llama3-70B	76.0	**97.6**	58.9	69.2	78.5
Mixtral-8x7b	77.8	95.0	64.0	73.4	78.7
Claude-Opus	80.7	94.7	60.3	**89.1**	78.7
Gem1.5-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
Gem1.5-Pro	**88.1**	92.3	**80.6**	87.5	92.0
(our 24b models)					
PALM-2-24B	62.9	89.9	61.2	55.3	45.2
FLAME	84.6	94.4	69.1	80.7	94.1
FLAME-RM	86.7	94.7	71.7	85.7	**94.8**

Table 2: A comparison of FLAME with other generative reward models ("LLM-as-judges") on the REWARDBENCH benchmark. FLAME outperforms all generative models on REWARDBENCH except proprietary Gemini 1.5 Pro, despite being trained only on public data.

rameters, which is necessary to evaluate ExpertQA answers. Surprisingly, we also find GPT-4-0125 generally outperforms GPT-4o on quality assessment tasks.[12] Finally, we note that FLAME variants outperform our initialization checkpoint (PALM-2 24B) on both tasks, showcasing the utility of FLAME fine-tuning.

Our FLAME-RM variant outperforms GPT-4 on RewardBench. In Table 2, we find that on average FLAME-RM outperforms several proprietary LLM-as-judge generative baselines on REWARDBENCH, including GPT-4-0125 (86.7 vs 85.9). This is due to a notable performance increase in the "Reasoning" split of the REWARDBENCH benchmark, with competitive performance in other splits. Moreover, FLAME-RM outperforms the much larger and open-source LLAMA-3-70B on every split of REWARDBENCH (86.7 vs 76.0) on average. Even our vanilla FLAME variant, without tail-patch optimization (§3.4), shows strong REWARDBENCH performance (84.6), outperforming models like Claude-Opus (80.7) and Gemini 1.5 Flash (82.1).[13]

5 Further analysis of FLAME

In this section, we provide an analysis to elucidate some interesting aspects of our models. We depart from the traditional focus on analyzing the effect of factors like model size, data size, and data quality within multitask learning, which have been extensively studied in recent work on multitask/instruction learning (Raffel et al., 2020; Longpre et al., 2023). Instead, we explore the biases inherent in these autoraters, and demonstrate their potential utility for AI development, such as sampling high-quality responses.

5.1 Autorater Bias Analysis

A common criticism of LLM-AS-A-JUDGE autoraters is bias towards certain judgments (Liu et al., 2023a; Panickssery et al., 2024). In this section, we evaluate FLAME on the CoBBLEr benchmark (Koo et al., 2023), and find that FLAME is significantly less biased than alternatives. This benchmark measures six kinds of biases in autorater models: (1) ORDER: does the autorater have a preference towards the response position? (2) COMPASSION: does the autorater's judgment change if the response-

[12]This comes as a surprise as GPT-4o is ranked higher than GPT-4-0125 on the LMSys leaderboard (Chiang et al., 2024). Our results corroborate to the REWARDBENCH leaderboard, where GPT-4o is ranked behind GPT-4-0125.

[13]We present some additional analysis of length bias and hill-climbing issues in REWARDBENCH in Appendix B. We encourage readers and future work to not over-index on REWARDBENCH performance, and instead consider holistic improvements across a variety of evaluation tasks (§4.1).

Autorater	Avg (↓)	Order (↓)	Compass. (↓)	Length (↓)	Egocentric (↓)	Bandwagon (↓)	Attention (↓)
Random	0.30	0.50	0.50	0.00	0.25	0.25	0.25
(baselines as reported in Koo et al., 2023)							
Falcon	0.31	0.77	0.27	0.09	**0.05**	0.28	0.40
Cohere	0.41	0.50	0.65	0.10	0.27	0.82	0.14
LLAMA2-70B	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-RM	0.15	0.14	0.20	0.03	0.35	0.20	**0.00**
FLAME	0.11	**0.08**	**0.12**	**0.00**	0.35	0.10	**0.00**

Table 3: Autorater bias analysis on the CoBBLEr benchmark from Koo et al. (2023). For all columns, **lower is better / less biased**. Overall, we find that **FLAME is significantly less biased** than popular LLM-as-a-judge models like GPT-4 and ChatGPT. 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 variant in Egocentric.
.

Ranker	CodeGen16B	davinci002	InCoder6B
(10 samples reranked in round-robin fashion)			
None	21.2	17.6	14.6
FLAME	31.7	22.0	**18.9**
FLAME-RM	**32.9**	**25.0**	**18.9**
Oracle	46.9	63.4	29.3

Table 4: Pass@1 performance on the HumanEval coding benchmark. Across models, ranking 10 samples with FLAME improves pass@1 performance, with FLAME-RM outperforming FLAME.

generating LLM's name is used instead of aliases? (3) LENGTH: does the autorater have a preference for longer or shorter outputs? (4) EGOCENTRIC: does the autorater have a preference for outputs generated by itself? (5) BANDWAGON: does the autorater get swayed by sentences like *"90% people prefer response A"*? (6) ATTENTION: does the autorater get distracted by irrelevant sentences about responses, such as *"Response A is about cats."*? We leverage the original prompt/response pairs from Koo et al. (2023), adapting them to use the unified FLAME format (Figure 1). We compare FLAME's bias to other LLM-as-judges reported in Koo et al. (2023), including GPT-4.

In Table 3, we find that **FLAME is significantly less biased than** GPT-4 and other autoraters reported in Koo et al. (2023), with an average bias of just 0.12 compared to 0.31 in GPT-4 (lower is better). FLAME outperforms GPT-4 in 5 out of 6 bias categories, further supporting its utility as a robust and unbiased autorater.

5.2 Using FLAME to re-rank decoded outputs

A possible application of autoraters is selecting the best output among a pool of responses (Nakano et al., 2021; Krishna et al., 2022), a technique popularly known as *"Best-of-N"* sampling. In this section, we show that ranking LLM-generated code samples with FLAME leads to performance improvements. We utilize the popular HumanEval Python programming benchmark (Chen et al., 2021) for our experiments. We re-rank 10 samples generated by OpenAI davinci-002, InCoder-6B (Fried et al., 2023), and CodeGen-16B (Nijkamp et al., 2023) using a round-robin competition, and measure the performance of the top-ranked sample.[14] In Table 4, we find that, **we can significantly improve pass@1 accuracy by ranking 10 output samples** for all three code-generation models. On CodeGen16B, FLAME-RM improves pass@1 from 21.2 to 32.9, bridging nearly half the gap to the Oracle ranker (46.9).

6 Conclusion

We introduce FLAME, a family of foundational autorater models that can perform various quality assessment tasks. FLAME is trained on a large and diverse collection of curated and standardized human evaluations derived exclusively from permissively licensed datasets. We demonstrate FLAME's strong zero-shot generalization abilities, outperforming proprietary models like GPT-4, CLAUDE on many held-out tasks. Additionally, we present a novel mixture weight tuning approach that dramatically improves effectiveness and efficiency on reward modeling. FLAME is the highest performing generative reward model trained only on permissively licensed data, and exhibits significantly less bias than popular LLM-AS-A-JUDGE models.

[14] We use relatively weak LLMs from Chen et al. (2023) since: (1) we want to study whether weaker LLMs can benefit from FLAME re-ranking; (2) HumanEval benchmark has been extensively hill-climbed on to develop newer 2024 LLMs.

Limitations and Future work

Our data collection faces challenges due to evolving model evaluation standards and the need for new evaluation types for emerging applications. Expanding our collection with open-source contributions could address this issue. Our models, trained primarily on English data with a context length of 2048 tokens, might not perform well on multilingual or long-context quality assessment tasks, such as book-length summarization evaluation. In future releases, we plan to include training on more multilingual datasets with longer context lengths. Finally, in this work, we train our models using a supervised multitask fashion. Exploring alternative training approaches like 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-Team 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, over-reliance 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.

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. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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 *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

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*, 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*.

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*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E

Gonzalez, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.

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*, 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*, 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, 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*, pages 9301–9323.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

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. *Advances in Neural Information Processing Systems*, 36.

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*, 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*, 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*, 10:1066–1083.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *International Conference on Machine Learning*, pages 5988–6008. PMLR.

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*, 9:391–409.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André FT Martins, Graham Neubig, Ankush Garg, Jonathan H 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*, pages 1066–1083.

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*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Yanjun Gao, Chen Sun, and Rebecca J Passonneau. 2019. Automated pyramid summarization evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*.

Gemini-Team, 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*.

Peiyuan Gong and Jiaxin Mao. 2023. Coascore: Chain-of-aspects prompting for nlg evaluation. *arXiv preprint arXiv:2312.10355*.

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 (Volume 1: Long Papers)*, pages 3785–3801.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. Q2: 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*, 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, Juntao 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 *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Dongfu Jiang, Yishan Li, Ge Zhang, Wenhao Huang, Bill Yuchen Lin, and Wenhu Chen. 2023. Tigerscore: Towards building explainable metric for all text generation tasks. *arXiv preprint arXiv:2310.00752*.

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*, 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*, pages 419–451.

Daniel Khashabi, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A Smith, and Daniel S Weld. 2021. Genie: Toward reproducible and standardized human evaluation for text generation. *arXiv preprint arXiv:2101.06561*.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491*.

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. *arXiv preprint arXiv:2405.01535*.

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

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*, volume 202 of *Proceedings of Machine Learning Research*, pages 17506–17533. PMLR.

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*, pages 1650–1669.

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*, 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*.

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. *Advances in Neural Information Processing Systems*, 36.

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, et al. 2021. Datasets: A community library for natural language processing. *arXiv preprint arXiv:2109.02846*.

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

Ruosen Li, Teerth Patel, and Xinya Du. 2023b. Prd: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv:2307.02762*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023c. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

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 Chi Kit 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*, 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*.

Bill Yuchen Lin, Khyathi Chandu, Faeze Brahman, Yuntian Deng, Abhilasha Ravichander, Valentina Pyatkin, Ronan Le Bras, and Yejin Choi. 2024. Wildbench: Benchmarking language models with challenging tasks from real users in the wild.

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 (Volume 1: Long Papers)*, pages 3214–3252.

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

Yixin Liu, Alexander R Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. 2023b. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. *arXiv preprint arXiv:2311.09184*.

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 (Volume 1: Long Papers)*, 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*, pages 9004–9017.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

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*, pages 12076–12100.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3470–3487.

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 Randy Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro Von Werra, and Shayne Longpre. 2024. Octopack: Instruction tuning code large language models. In *The Twelfth International Conference on Learning Representations*.

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*.

OpenAI. 2024a. GPT-3.5 Turbo.

OpenAI. 2024b. 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*, 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*.

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*, 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. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Yiwei Qin, Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2023. T5score: Discriminative fine-tuning of generative evaluation metrics. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15185–15202.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 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)*, 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, et al. 2023. Scaling up models and data with t5x and seqio. *Journal of Machine Learning Research*, 24(377):1–8.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. *Proceedings of the 10th 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*, 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*, pages 7881–7892, Online. Association for Computational Linguistics.

Ondrej Skopek, Rahul Aralikatte, Sian Gooding, and Victor Cărbune. 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. *Advances in Neural Information Processing Systems*, 33:3008–3021.

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*, pages 5008–5020.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, et al. 2023. 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*, 7:625–641.

Jason Wei, Maarten Bosma, Vincent Y 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. *Proceedings of the 10th 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 (Long Papers)*, 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 *Thirty-seventh Conference on Neural Information Processing Systems*.

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 (Volume 1: Long Papers)*, 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 (Volume 1: Long Papers)*, 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*, 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 (Volume 1: Long Papers)*, pages 10457–10480.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Process-*

ing Systems, volume 34, pages 27263–27277. Curran Associates, Inc.

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*.

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 (Long and Short Papers)*, pages 1298–1308.

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. 2023a. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

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. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Appendix

A Appendix Tables

A.1 List of Training Datasets in FLAME

Please see Table 6.

B Further analysis on the RewardBench benchmark

In this section, we provide some analysis of issues we found in the RewardBench benchmark, including issues of length bias (Appendix B.1) and difficulty in hill-climbing (Appendix B.2). Given these issues, we encourage readers and future efforts to not solely rely on RewardBench for autorater task performance, but instead use a wide variety of evaluation tasks to compare models (such as our evaluation suite in §4).

B.1 Analysis of length and token bias in RewardBench

In Table 5, we present an analysis of length bias in RewardBench. Overall, we find that RewardBench is far from length-balanced in its composition. The Chat-Hard, Coding and Math categories strongly prefer shorter length outputs, while the Chat category has a strong preference towards longer outputs. An adversarial submission which can identify prompt categories may simply choose to prefer the longer or shorter output in RewardBench, and achieve high scores without being an actually strong preference model.

RewardBench Split	% longer preferred
Chat	79.1%
Chat-Hard	29.6%
Math	6.5%
Coding	35.7%
Safety	41.9%

Table 5: A summary of length bias in RewardBench. Overall, we find that four out of five categories in RewardBench have a strong preference towards longer or shorter outputs.

Besides length bias, we also found some issues of token bias in the Math and Safety splits of RewardBench. In Safety, the preferred side had a strong preference for phrases like *"I'm sorry"*, which are indicative of hedged responses. In 28% pairs, only the preferred response contained the word "sorry". We found similar issues in the Math split, with tokens "i", "can", "need", "to", "find" largely only appearing in the rejected response.

Given these findings, our recommendation for future autorater / reward model development is **not just rely on REWARDBENCH performance, but instead to evaluate a wide variety of autorater tasks** (such as our evaluation suite in §4).

B.2 Discussion on hill-climbing on REWARDBENCH

In early experiments, we found it very difficult to hill-climb on REWARDBENCH due to the absence of a development set in Lambert et al. (2024). It was not possible to construct a "proxy" development set for many categories of REWARDBENCH, since Lambert et al. (2024) had fully utilized the original set while constructing them (such as LLMBAR in CHATHARD). In early experiments, we saw poor correlation between REWARDBENCH performance and performance on other held-out tasks between our model variants. This prompted us to hill-climb directly on REWARDBENCH as a proxy, as described in §3.4. To confirm we have not overfit on REWARDBENCH, we evaluate both FLAME and FLAME-RM on several other held-out tasks besides REWARDBENCH, to confirm the general purpose utility of our models across a variety of tasks.

Capability	Dataset	Source	Output Format
Attribution / Factuality	ESNLI	Camburu et al. (2018)	Classification, Generative
	MNLI	Williams et al. (2018)	Classification
	VitaminC	Schuster et al. (2021)	Classification
	Sentence Similarity - CxC	Parekh et al. (2021)	Pointwise
	Sentence Similarity - STSB	Cer et al. (2017)	Pointwise
	MultiPIT	Dou et al. (2022b)	Classification
	QQP	Iyer et al. (2017)	Classification
	PAWS Paraphrasing	Zhang et al. (2019)	Classification
	FaithDial	Dziri et al. (2022a)	Classification
	MOCHA	Chen et al. (2020)	Pointwise
	DialFact	Gupta et al. (2022)	Classification
	RAGTruth	Wu et al. (2023a)	Classification
	FActScore	Min et al. (2023)	Classification
	FRANK	Pagnoni et al. (2021)	Classification
	BEGIN	Dziri et al. (2022b)	Classification
	XSUM-Faithful	Maynez et al. (2020)	Generative
	HaluEval	Li et al. (2023a)	Classification
	QAGS	Wang et al. (2020)	Classification
	WikiBio Hallucinations	Manakul et al. (2023)	Pointwise
	Q2	Honovich et al. (2021)	Classification
General Text Quality	GENIE	Khashabi et al. (2021)	Pointwise, Pairwise, Gen.
	InstruSum	Liu et al. (2023b)	Pairwise, Classification
	RiSum	Skopek et al. (2023)	Pointwise, Classification
	Stanford SHP	Ethayarajh et al. (2022)	Pairwise
	BeaverTails Helpful	Ji et al. (2023)	Pairwise
	HH RLHF Helpful	Bai et al. (2022)	Pairwise
	Summary Feedback Comparisons	Stiennon et al. (2020)	Pairwise, Pointwise
	SEAHORSE	Clark et al. (2023)	Classification
	Scarecrow	Dou et al. (2022a)	Classification, Generative
	SummaEval	Fabbri et al. (2021)	Pointwise
	LMSys Chatbot Arena (english)	Zheng et al. (2023a)	Pairwise
	FeedbackQA	Li et al. (2022b)	Pointwise, Generative
	WebGPT	Nakano et al. (2021)	Pairwise, Generative
	Fine-grained RLHF	Wu et al. (2023b)	Pairwise, Classification
	LENS	Maddela et al. (2023)	Pointwise
	MAUVE - Human Eval	Pillutla et al. (2021)	Pairwise
	CoLA	Warstadt et al. (2019)	Classification
	CREPE	Yu et al. (2023)	Classification, Generative
	PRD-Vicuna	Li et al. (2023b)	Pairwise
	Hurdles LFQA	Krishna et al. (2021)	Pairwise
	Validity LFQA	Xu et al. (2022)	Classification, Generative
	News Summarization Evaluation	Goyal et al. (2022)	Pairwise
	Helpful Steer (training split)	(Wang et al., 2023)	Pointwise
Safety	BeaverTails Classify	Ji et al. (2023)	Classification
	BeaverTails Harmless	Ji et al. (2023)	Pairwise
	HH RLHF Harmless	Bai et al. (2022)	Pairwise
	HH RLHF Red Teaming	Bai et al. (2022)	Pointwise
Coding	Code Contests	Li et al. (2022a)	Pairwise
	COFFEE	Moon et al. (2023)	Pairwise
	CommitPack	Muennighoff et al. (2024)	Pairwise
	CommitPack - Bugs	Muennighoff et al. (2024)	Pairwise
Math Reasoning	PRM800K preference	Lightman et al. (2024)	Pairwise
Instruction Tuning	TULU V2	Ivison et al. (2023)	Generative
	PRM800K-IF	Lightman et al. (2024)	Generative
	LIMA	Zhou et al. (2023)	Generative

Table 6: A complete list of datasets which were used to train FLAME, along with their output format and capability categorization. From many source datasets, we derived multiple training dataset tasks (for example, by splitting the pointwise and pairwise ratings for the the same set of responses).