

# PROMETHEUS 2: An Open Source Language Model Specialized in Evaluating Other Language Models

Seungone Kim<sup>1,2,3\*</sup> Juyoung Suk<sup>1\*</sup> Shayne Longpre<sup>4</sup> Bill Yuchen Lin<sup>5</sup> Jamin Shin<sup>1</sup>  
Sean Welleck<sup>3</sup> Graham Neubig<sup>3</sup> Moontae Lee<sup>2,6</sup> Kyungjae Lee<sup>2</sup> Minjoon Seo<sup>1</sup>

KAIST AI<sup>1</sup> LG AI Research<sup>2</sup> Carnegie Mellon University<sup>3</sup> MIT<sup>4</sup>  
Allen Institute for AI<sup>5</sup> University of Illinois Chicago<sup>6</sup>  
seungone@cmu.edu {juyoung, minjoon}@kaist.ac.kr

## Abstract

Proprietary LMs such as GPT-4 are often employed to assess the quality of responses from various LMs. However, concerns including transparency, controllability, and affordability strongly motivate the development of open-source LMs specialized in evaluations. On the other hand, existing open evaluator LMs exhibit critical shortcomings: 1) they issue scores that significantly diverge from those assigned by humans, and 2) they lack the flexibility to perform both direct assessment and pairwise ranking, the two most prevalent forms of assessment. Additionally, they often do not possess the ability to evaluate based on *custom evaluation criteria*, focusing instead on general attributes like helpfulness and harmlessness. To address these issues, we introduce Prometheus 2. Prometheus 2 is more powerful than its predecessor, and closely mirrors human and GPT-4 judgements. Moreover, it is capable of processing both direct assessment and pair-wise ranking formats grouped with a user-defined evaluation criteria. On four direct assessment benchmarks and four pairwise ranking benchmarks, PROMETHEUS 2 scores the highest correlation and agreement with humans and proprietary LM judges among all tested open evaluator LMs. Our models, code, and data are all publicly available.<sup>1</sup>

## 1 Introduction

Evaluating the quality of outputs produced by language models (LMs) is progressively becoming difficult, as the outputs cover an extremely diverse distribution of text and complex tasks. To address this issue, language model-based evaluation has emerged as a scalable and cheap paradigm for assessing LM-generated text (Li et al., 2024;

\*equal contribution. Work was done while Seungone was an intern at LG AI Research and a MS student at KAIST.

<sup>1</sup><https://github.com/prometheus-eval/prometheus-eval>

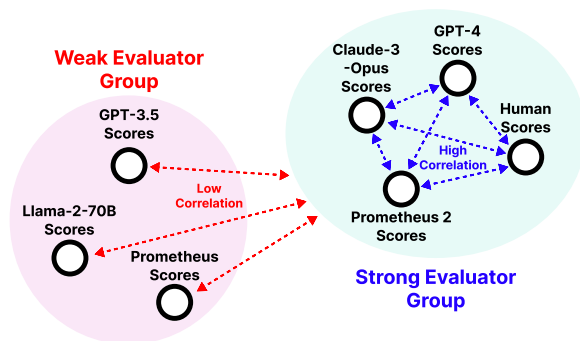


Figure 1: Weak evaluators (e.g., Llama-2-Chat-70B, Prometheus, and GPT-3.5-Turbo) achieve low scoring correlation with strong evaluators (e.g., Humans, GPT-4, and Claude-3-Opus). On the other hand, scores provided by strong evaluators highly correlate with each other.

Gao et al., 2024). In this paradigm, LMs are either prompted to output a scalar indicator of quality (denoted as *direct assessment*) (Zheng et al., 2023; Liu et al., 2023b; Ye et al., 2023; Kim et al., 2023) or to determine which of two outputs are preferred (denoted as *pairwise ranking*) (Wang et al., 2023b; Li et al., 2023b; Lambert et al., 2024). Prior works employing proprietary LMs as evaluators have demonstrated not only high correlations with human evaluations but also increased speed and cost-effectiveness (Zheng et al., 2023; Liu et al., 2023b; Dubois et al., 2023; Ye et al., 2023).

However, relying on proprietary LMs for evaluation poses significant challenges. The lack of transparency about their training data compromises both fairness and reproducibility, making it problematic to use them in evaluation pipelines. Additionally, concerns regarding controllability and affordability also persist (Kim et al., 2023). To address these issues, recent works have focused on developing evaluator LMs that are open-access, transparent, and controllable (Kim et al., 2023; Wang et al., 2023a,b; Li et al., 2023a; Zhu et al., 2023; Jiang et al., 2023b,c; Lee et al., 2024). Yet, these models often yield scoring decisions that do not correlate well enough with human judgments or those made

by proprietary LMs, failing to effectively simulate them. Moreover, open evaluator LMs are not flexible since they are typically trained only to perform either direct assessment or pairwise ranking and assess based on general public preferences like helpfulness and harmlessness, limiting their ability to handle diverse real-life scenarios.

To close the gap with proprietary LMs, we investigate *unifying* the two model-based evaluation paradigms - direct assessment and pairwise ranking - to train a robust unified evaluator LM. We propose a recipe based on merging the weights of two evaluator LMs trained separately on direct assessment and pairwise ranking formats. Our key empirical observation is that weight merging can yield an evaluator LM that not only *works* in both formats, but also *outperforms* evaluator LMs that are jointly trained or only trained on a single format.

To demonstrate our approach, we develop the PREFERENCE COLLECTION, a new fine-grained pairwise ranking feedback dataset that builds on the FEEDBACK COLLECTION (Kim et al., 2023), which is a direct assessment feedback dataset. We choose Mistral-7B (Jiang et al., 2023a) and Mixtral-8x7B (Jiang et al., 2024) as our base models, and merge the weights of evaluator LMs separately trained on the FEEDBACK COLLECTION and the PREFERENCE COLLECTION to obtain our resulting models, PROMETHEUS 2 (7B & 8x7B).

On four direct assessment benchmarks (Vicuna Bench, MT Bench, FLASK, Feedback Bench), the PROMETHEUS 2 models demonstrate the highest correlation with both human evaluators and proprietary LM-based judges compared to existing open evaluator LMs, with the Pearson correlation surpassing other baselines by 0.2 units across all datasets. Similarly, on four pairwise ranking benchmarks (HHH Alignment, MT Bench Human Judgment, Auto-J Eval, Preference Bench), the PROMETHEUS 2 models show the highest agreement with human evaluators among all the open evaluator LMs we tested, reducing the performance gap with GPT-4 in half.

Our contributions are summarized as follows:

- We introduce PROMETHEUS 2 (7B & 8x7B), state-of-the-art open evaluator LMs that score high correlations with both human evaluators and proprietary LM-based judges on both direct assessment and pairwise ranking.
- We introduce a pairwise ranking feedback dataset called the PREFERENCE COLLEC-

TION, which includes 1K custom evaluation criteria beyond helpfulness and harmlessness.

- We show that merging the weights of evaluator LMs trained on direct assessment and pairwise ranking feedback datasets results in a unified evaluator LM that excels in both schemes.

## 2 Related Work

### 2.1 Language Model-based Evaluation

To assess the generation capabilities of LMs, prior works such as the GEM benchmark (Gehrmann et al., 2021, 2022) employed ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), and BERTScore (Zhang et al., 2019) as their metrics, which measure the lexical or semantic similarity between a reference answer and a response. However, these conventional metrics are prone to false negatives because they are not expressive enough to recognize responses that are of good quality but differ from the reference answer (Schluter, 2017; Freitag et al., 2020; Hanna and Bojar, 2021).

Recently, employing language models as a judge has gained attention as a promising paradigm to mimic the depth and granularity that human evaluation offers (Zheng et al., 2023; Liu et al., 2023b; Li et al., 2023b; Chan et al., 2023; Ye et al., 2023). To reduce the over-reliance on proprietary LMs, follow-up works suggest training language models specialized in evaluations (Cui et al., 2023; Kim et al., 2023; Jiang et al., 2023b,c; Li et al., 2023a; Lee et al., 2024). Yet, open evaluator LMs do not possess the flexibility to function in different evaluation schemes and show weak evaluation performance compared to proprietary LMs. We aim to bridge this gap by introducing PROMETHEUS 2.

### 2.2 Weight Merging

Prior works have demonstrated that weight merging can enhance performance across various domains, including language modeling (Li et al., 2022; Matena and Raffel, 2022; Ilharco et al., 2022; Don-Yehiya et al., 2022; Gururangan et al., 2023; Yadav et al., 2024; Sukhbaatar et al., 2024), instruction-tuning (Jang et al., 2023b; Yu et al., 2023), and aligning to user preferences (Jang et al., 2023a; Rame et al., 2024; Wang et al., 2024). In our work, we specifically focus on enhancing the evaluation capabilities of open evaluator LMs. By merging models trained on different assessment formats—specifically, direct assessment and pairwise

ranking—we aim to obtain an evaluator LM that not only functions in both formats but also shows as good evaluation performances as proprietary LMs.

### 3 Methodology

We propose a new recipe for training a unified evaluator LM based on merging the weights of models trained for direct assessment and pairwise ranking. We begin with background on direct assessment and pairwise ranking for evaluator LMs (Section 3.1, 3.2), followed by the construction process of our training data (Section 3.3). Finally, we present our methods to train state-of-the-art evaluator LMs, Prometheus 2 models (Section 3.4).

#### 3.1 Direct Assessment

Direct assessment is mapping an instruction  $i$  and response  $r$  into a scalar value score  $s$ , such as  $f_{direct} : (i, r) \mapsto s$  where  $s \in \mathbb{R}$ . For the scoring range, we use an integer between 1 and 5.

Prior works have identified several recipes to align the scores provided by evaluator LMs ( $s_{LM}$ ) and the scores assigned by humans ( $s_{human}$ ). For instance, Liu et al. (2023a) and Zheng et al. (2023) have shown that it is crucial to add a reference answer  $a$  as input to the evaluator LM to maximize the correlation between  $s_{LM}$  and  $s_{human}$ . Also, Zheng et al. (2023) and Ye et al. (2023) showed that prompting the language model to write verbal feedback  $v_r$  before  $s$  also improves the correlation between  $s_{LM}$  and  $s_{human}$ . Lastly, Ye et al. (2023) and Kim et al. (2023) showed that by explicitly integrating evaluation criteria  $e$ , users can define the standards for model assessment, ensuring evaluations are flexible to specific needs rather than generic qualities. Specifically,  $e$  is represented as a score rubric including a description for the criterion itself and a set of descriptions for each score between the scoring range. This is expressed as:

$$f_{direct} : (i, r, a, e) \mapsto (v_r, s) \quad (1)$$

where  $s \in \{1, 2, 3, 4, 5\}$

#### 3.2 Pairwise Ranking

Pairwise ranking is mapping an instruction  $i$  and two pair of responses ( $r_m, r_n$ ) into either  $i$  or  $j$ , such as  $f_{pair} : (i, r_m, r_n) \mapsto s$  where  $s \in \{m, n\}$ .

Similar to direct assessment, prior works have identified that integrating a reference answer  $a$  and verbal feedback  $v_{r_m, r_n}$  into the evaluation pipeline is crucial (Zheng et al., 2023; Li et al., 2023b,a). In addition, to support granular assessment under

Data	PREFERENCE COLLECTION	FEEDBACK COLLECTION
<b>Evaluation Scheme</b>	Pairwise Ranking	Direct Assessment
<b># Evaluation Criteria</b>	1,000	1,000
<b># Instructions</b>	20,000	20,000
<b># Reference Answer</b>	20,000	20,000
<b># Instances</b>	200,000	100,000
<b># Verbal Feedback</b>	200,000	100,000

Table 1: Statistics of our training datasets, the FEEDBACK COLLECTION and the PREFERENCE COLLECTION. Note that the 1K evaluation criteria, 20K instructions, and 20K reference answers are *shared* among the two datasets. Both datasets have an equal number of scoring decisions (“A” or “B”; 100K each & 1-5; 20K each) to prevent unintended biases after training.

custom criterion, we add the evaluation criteria  $e$  as input to the evaluator LM (Ye et al., 2023; Kim et al., 2023). To the best of our knowledge, we are the first to study such fine-grained evaluation in pairwise ranking settings. This is expressed as:

$$f_{pair} : (i, r_m, r_n, a, e) \mapsto (v_{r_m, r_n}, s) \quad (2)$$

where  $s \in \{m, n\}$

In pairwise ranking, the evaluation criterion  $e$  does not include a set of descriptions for each score; instead, only the description of the evaluation criterion itself. Also, it is noteworthy that the verbal feedback  $v_{r_m, r_n}$  compares the commonalities and differences between  $r_m$  and  $r_n$  concerning  $e$ .

#### 3.3 The Preference Collection

Popular pairwise ranking datasets such as HH-RLHF (Bai et al., 2022) or Ultra Feedback (Cui et al., 2023) do not include an evaluation criterion  $e$  and a verbal feedback  $v_{r_m, r_n}$ . To train an evaluator LM that could assess based on such criteria, we construct the PREFERENCE COLLECTION, including 1K evaluation criteria. We apply two modifications to the FEEDBACK COLLECTION. First, since the FEEDBACK COLLECTION includes five responses for each instruction, each corresponding to a scoring decision between 1 and 5, we pair two out of the five responses, resulting in a total of ten combinations per instruction. Using the existing scoring decisions for each response, we determine which response is better and assign a new scoring decision for that pair (*i.e.*, “Response A is better” or “Response B is better”). Second, to generate new verbal feedback  $v_{r_m, r_n}$  for each pair of responses, we prompt GPT-4-1106 to identify the commonalities and differences between the two responses.

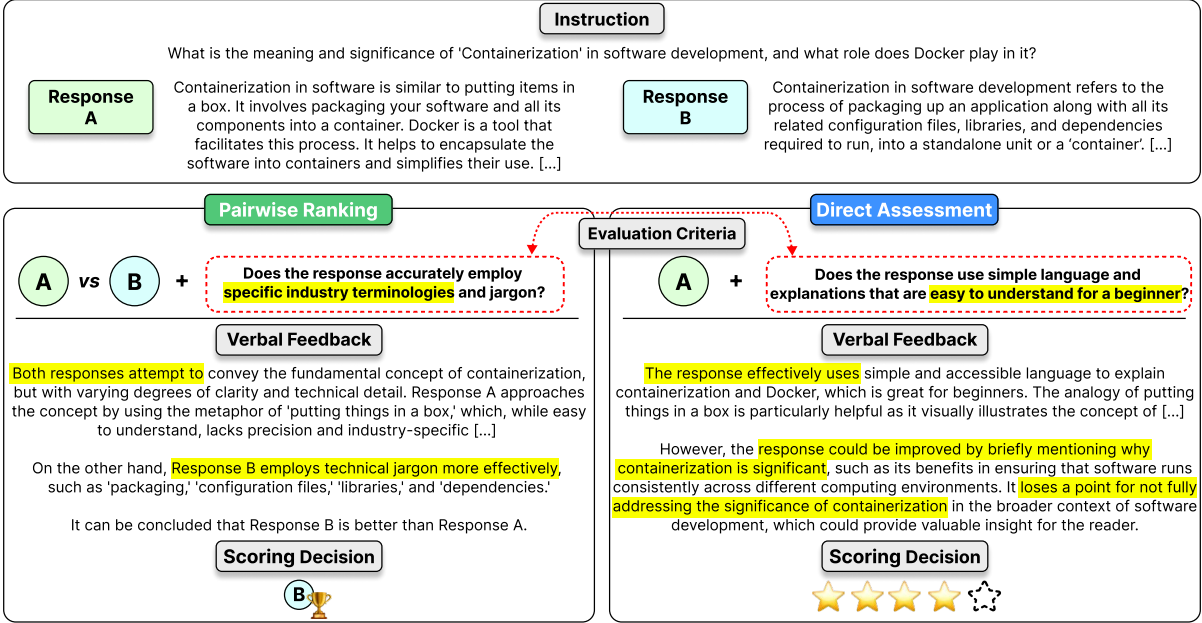


Figure 2: Comparison of direct assessment and pairwise ranking. Both responses could be considered decent under the umbrella of ‘helpfulness’. However, the scoring decision might change based on a specific evaluation criterion.

The statistics of the resulting dataset are listed in Table 1 along with the FEEDBACK COLLECTION. We explain about our quality verification process of the PREFERENCE COLLECTION in Appendix A. Also, we include the prompts we use for the augmentation process in Appendix H.

### 3.4 Training Methods & Baselines

**Prompting** Prompting involves querying an LM to make judgments in a specified evaluation format without training. We employ Llama-2-Chat-7,13,70B (Touvron et al., 2023); Mistral-7B-Instruct-v0.2 (Jiang et al., 2023a); and Mixtral-8x7B-Instruct-v0.1 (Jiang et al., 2024) as our baselines. It’s worth noting that models not explicitly trained on feedback data often fail to generate responses in the required format, making it extremely difficult to parse scoring decisions. Although it is impractical for regular use, we make a fair comparison by infinitely looping until scores can be parsed. Also, we include proprietary LMs such as GPT-3.5-Turbo-0613; GPT-4-1106; and Claude-3-Opus.

**Single-Format Training** Single-Format training involves training a base model  $\theta$  on either on a direct assessment feedback dataset  $D_d$  or a pairwise ranking feedback dataset  $D_p$ . For single-format trained evaluator LMs, we test Prometheus-7,13B (Kim et al., 2023) (direct assessment); UltraRM-13B (Cui et al., 2023) (pairwise ranking); and PairRM-0.4B (Jiang et al., 2023c) (pair-

wise ranking). In addition, we also report the performances of single-format training Mistral-7B-Instruct-v0.2 and Mixtral-8x7B-Instruct-v0.1 on either direct assessment or pairwise ranking.

**Joint Training** Joint training involves training a base model  $\theta$  on both a direct assessment feedback dataset  $D_d$  and a pairwise ranking feedback dataset  $D_p$ . This enables the resulting evaluator LM to function across both evaluation formats. For jointly trained evaluator LMs, we test Auto-J (Li et al., 2023a). In addition, we report the performances of jointly training Mistral-7B and Mixtral-8x7B on both direct assessment and pairwise ranking.

**Weight Merging** Weight Merging involves training two models,  $\theta_d$  and  $\theta_p$ , separately on a direct assessment feedback dataset  $D_d$  and a pairwise ranking feedback dataset  $D_p$ . Then, the final evaluator LM  $\theta_{final}$  is obtained by merging  $\theta_d$  and  $\theta_p$ . For example, linear merging is as follows:

$$\theta_{final} = \alpha \times \theta_d + (1 - \alpha) \times \theta_p \quad (3)$$

In addition to linear merging, we test 5 additional variants, namely Task Arithmetic merging (Ilharco et al., 2022), TIES merging (Yadav et al., 2024), DARE-TIES and DARE-Linear merging (Yu et al., 2023), and SLERP merging (Goddard et al., 2024). We include an explanation of these merging methods and ablation experiment results of the performance differences in Appendix G. Among them,

Evaluation Method	Benchmark	Metrics	Judgment Source	Reference Answer	# Score Rubrics	# Instructions	# Judgments
Direct Assessment	Vicuna Bench	Correlation	Proprietary LMs	Y	80	80	320
	MT Bench	Correlation	Proprietary LMs	Y	80	80	320
	FLASK	Correlation	Proprietary LMs & Humans	Y	12	200	2,000
	Feedback Bench	Correlation	Proprietary LMs	Y	200	200	1,000
Pairwise Ranking	HHH Align.	Accuracy	Humans	N	4	221	221
	MT Bench Human Judg.	Accuracy	Humans	N	1	80	3,360
	Auto-J Eval	Accuracy	Humans	N	1	58	1,392
	Preference Bench	Accuracy	Proprietary LMs	Y	200	200	2,000

Table 2: Statistics of our evaluation benchmarks to assess the evaluation capabilities of evaluator LMs.

DARE-Linear showed the best performance, and hence we used it to train the PROMETHEUS 2 (7B & 8x7B) models. Details on the hyper-parameters for training and inference along with the prompt templates are all listed in Appendix B, I, J.

## 4 Experimental Setup

The statistics of all the benchmarks are in Table 2. The four direct assessment benchmarks are:

- **Vicuna Bench** (Chiang et al., 2023): A single-turn chat benchmark that includes 80 test prompts, 80 hand-crafted score rubrics from Kim et al. (2023), and 320 responses obtained by WizardLM-13B, Vicuna-13B, Llama-2-Chat-13B, GPT-3.5-Turbo-0613.
- **MT Bench** (Zheng et al., 2023): A multi-turn chat benchmark that consists of 80 test prompts, 80 hand-crafted score rubrics from Kim et al. (2023), and 320 responses obtained by WizardLM-13B, Vicuna-13B, Llama-2-Chat-13B, GPT-3.5-Turbo-0613.
- **FLASK** (Ye et al., 2023): A fine-grained evaluation benchmark comprised of 200 test prompts, 12 score rubrics, and 2000 responses acquired from Alpaca-7B, Vicuna-13B, Bard, GPT-3.5-Turbo-0613. In addition to scores from proprietary LMs, this benchmark also includes scores marked by human evaluators.
- **Feedback Bench** (Kim et al., 2023): The test set of the FEEDBACK COLLECTION with 1K score rubrics, 200 instructions, and 1K responses that do not overlap with the train data.

The four pairwise ranking benchmarks are:

- **HHH Alignment** (Askeff et al., 2021): A benchmark consisting of 221 prompts; 4 score rubrics (helpfulness, harmlessness, honesty, and other) and 221 response pairs (graded as ‘win’ or ‘lose’) judged by human evaluators.

- **MT Bench Human Judgment** (Zheng et al., 2023): A benchmark that shares the same 80 prompts as MT-Bench. In addition, it provides 3,360 response pairs (graded as ‘win’, ‘tie’, or ‘lose’) judged by human evaluators.
- **Auto-J Eval** (Li et al., 2023a): A benchmark consisted of 58 prompts and 1,392 response pairs (graded as ‘win’, ‘tie’, or ‘lose’) judged by human evaluators. This benchmark is used as the in-domain test set of Auto-J.
- **Preference Bench**: Our in-domain test set for the PROMETHEUS models. Similar to how the PREFERENCE COLLECTION was made with the FEEDBACK COLLECTION, we adjust the FEEDBACK BENCH and pair two out of the five responses, resulting in a test set with 200 prompts, 2,000 response pairs (graded as ‘win’ or ‘lose’), and 200 evaluation criteria.

In direct assessment, we conduct **reference-based** evaluations by appending the reference answer as the input. We use **Pearson**, **Spearman**, and **Kendall-Tau** as performance metrics to measure scoring correlations against reference evaluators. Moreover, we include the results of the reference-free direct assessment evaluation in Appendix F.

In pairwise ranking, we conduct **reference-free** evaluations. Based on judgments assigned by humans, we use **accuracy** as our metric to measure agreement between evaluator LMs and humans.

Also, the MT Bench Human Judgment and Auto-J test set includes a ‘tie’ option assessed by human evaluators. We evaluate in two ways: by excluding all ‘tie’ options for pairwise ranking (denoted as ‘w/o tie’), or by using direct assessment where responses scored as ‘ties’ are grouped, and pairwise rankings are applied to the remaining responses with differing scores (denoted as ‘w/ tie’).

Evaluator LM	VICUNA BENCH		MT BENCH		FLASK			Feedback Bench
	GPT-4-1106	Claude-3-Opus	GPT-4-1106	Claude-3-Opus	GPT-4-1106	Claude-3-Opus	Humans	GPT-4-0613
LLAMA2-CHAT 7B	0.205	0.243	0.036	0.055	0.317	0.256	0.299	0.523
LLAMA2-CHAT 13B	0.185	0.141	-0.042	-0.002	0.239	0.247	0.263	0.545
LLAMA2-CHAT 70B	0.350	0.463	0.178	0.228	0.388	0.402	0.317	0.592
MISTRAL-INSTRUCT-7B	0.486	0.561	0.284	0.396	0.448	0.437	0.377	0.586
MIXTRAL-INSTRUCT-8x7B	0.566	0.579	0.551	0.539	0.483	0.495	0.420	0.673
PROMETHEUS-7B	0.484	0.528	0.378	0.382	0.352	0.331	0.348	0.847
PROMETHEUS-13B	0.492	0.534	0.404	0.477	0.462	0.470	0.449	0.860
AUTO-J (13B)	0.351	0.262	0.432	0.375	0.430	0.370	0.473	0.637
PROMETHEUS-2-7B	<u>0.666</u>	<b>0.654</b>	<u>0.548</u>	<u>0.517</u>	<u>0.617</u>	<u>0.561</u>	<u>0.545</u>	<u>0.882</u>
PROMETHEUS-2-8x7B	<b>0.685</b>	<u>0.635</u>	<b>0.665</b>	<b>0.614</b>	<b>0.659</b>	<b>0.626</b>	<b>0.555</b>	<b>0.898</b>
GPT-3.5-TURBO-0613	0.335	0.349	0.183	0.194	0.437	0.396	0.450	0.594
GPT-4-1106	/	0.694	/	0.717	/	0.736	0.679	0.753
CLAUDE-3-OPUS	0.694	/	0.717	/	0.736	/	0.573	0.788

Table 3: **Direct Assessment Results** Pearson correlations between reference evaluators (listed on top) and evaluator LMs. The best comparable statistics are **bolded** and second best underlined except proprietary LMs. Spearman and Kendall-Tau correlations are reported in Appendix C. Note that the Feedback Bench is an in-domain test set of the PROMETHEUS models.

Evaluator LM	HHH ALIGNMENT					MT BENCH HUMAN JUDG.		AUTO-J EVAL		Preference Bench
	Help.	Harm.	Hon.	Other	Total Avg.	w/ TIE	w/o TIE	w/ TIE	w/o TIE	Instance-wise Criteria
LLAMA2-CHAT 7B	55.93	62.07	49.18	62.79	57.01	46.68	50.39	45.76	45.73	58.60
LLAMA2-CHAT 13B	71.19	77.59	60.66	62.79	68.33	51.22	49.61	47.84	43.28	63.00
LLAMA2-CHAT 70B	62.71	81.03	65.57	65.12	68.78	<b>55.14</b>	60.88	53.38	50.64	64.70
MISTRAL-INSTRUCT-7B	59.32	68.97	63.93	81.40	67.42	53.81	63.82	53.88	60.94	79.40
MIXTRAL-INSTRUCT-8x7B	83.05	<u>87.93</u>	67.21	69.77	77.38	51.85	<u>71.42</u>	53.81	73.50	84.00
PAIR RM (0.4B)	<u>84.75</u>	84.48	<u>80.33</u>	<b>90.70</b>	<u>84.62</u>	-	59.00	-	59.05	81.80
ULTRA RM (13B)	<b>86.44</b>	79.31	<b>81.97</b>	<u>88.37</u>	83.71	-	56.00	-	59.85	86.97
AUTO-J (13B)	77.97	79.31	70.49	74.42	75.57	42.56	69.12	43.46	<u>76.64</u>	81.35
PROMETHEUS-2-7B	72.78	79.31	77.05	76.74	74.66	50.45	70.78	<u>54.96</u>	75.07	<b>93.25</b>
PROMETHEUS-2-8x7B	<u>84.75</u>	<b>96.55</b>	<b>81.97</b>	76.74	<b>85.52</b>	<u>55.07</u>	<b>71.96</b>	<b>58.41</b>	<b>79.98</b>	<u>90.65</u>
GPT-3.5-TURBO-0613	77.97	81.03	77.05	67.44	76.47	54.65	69.41	45.98	72.13	75.05
GPT-4-1106-PREVIEW	89.83	96.55	91.80	83.72	90.95	60.38	79.90	52.80	83.12	85.50
CLAUDE-3-OPUS	91.53	100.00	91.80	95.35	94.57	55.35	77.65	60.70	82.92	89.85

Table 4: **Pairwise Ranking Results** Accuracy on human preference datasets. The best comparable accuracies are **bolded** and second best underlined except proprietary LMs. Note that HHH Alignment is an in-domain test set for PairRM, Auto-J Eval is an in-domain test set for Auto-J, and the Preference Bench is an in-domain test set for Prometheus-2 models.

## 5 Experimental Results

In this section, we compare the evaluation capabilities of PROMETHEUS-2 models with other baselines using a direct assessment format (Section 5.1) and a pairwise ranking format (Section 5.2). Additionally, we measure the consistency of the scores from evaluator LMs in Appendix E.

### 5.1 Direct Assessment Results

The direct assessment results are shown in Table 3. The scoring decisions of PROMETHEUS 2 models (7B & 8x7B), GPT-4-1106, Claude-3-Opus, and human evaluators all strongly correlate with each other, yielding Pearson correlations higher than 0.5 regardless of the reference evaluator and benchmark. On the other hand, base LMs, single-format trained LMs, and jointly trained LMs show lower correlations, mostly falling below 0.5.

Notably, PROMETHEUS 2 models outperform Prometheus and Auto-J by at least 0.2 units across benchmarks in their correlation with proprietary LMs. Moreover, on the FLASK benchmark, while the correlation between humans and GPT-4 is 0.679, the highest correlation previously achieved by Prometheus-13B with humans was 0.449. PROMETHEUS-2-8x7B achieves a correlation of 0.555 with humans, halving the gap.

### 5.2 Pairwise Ranking Results

The pairwise ranking results are shown in Table 4. We exclude the results of Pair RM and Ultra RM on ‘w/ Tie’ settings since they could not process it.

On all of the 4 benchmarks, the PROMETHEUS 2 models achieve the highest scores, showing that they could effectively simulate human judgments. Notably, while HHH Alignment is an in-domain test set for Pair RM, and Auto-J Eval is for Auto-

J, PROMETHEUS-2-8X7B achieves higher scores. This shows that training a large LM (*i.e.*, Mixtral-8x7B) with feedback data could be an effective strategy to obtain a robust evaluator LM that could generalize beyond its training data. Moreover, the PROMETHEUS 2 models at least halve the performance gap with proprietary LMs compared to existing evaluator LMs on out-of-domain test sets.

## 6 Analyses of Weight Merging

To understand the effectiveness of our proposed weight merging method in the context of evaluations, we address the following research questions:

- **RQ1:** Is weight merging more effective compared to joint training? (Section 6.1)
- **RQ2:** Is the effectiveness of weight merging due to model ensembling? (Section 6.2)
- **RQ3:** To what extent does learning with direct assessment help pairwise ranking performance, and vice versa? (Section 6.3)

### 6.1 Weight Merging vs Joint Training

Table 5 compares the performance of evaluator LMs trained via weight merging and joint training. Alongside this, we also add and compare the results of prompting and single-format training.

Surprisingly, evaluator LMs trained via joint training often show lower performance compared to those trained only in single-format, which indicates *negative task transfer*. Specifically, evaluator LMs trained only on direct assessment formats obtain higher correlations compared to their jointly trained counterparts across different model scales. Similarly, evaluator LMs trained solely on pairwise ranking formats achieve higher average accuracy compared to those trained on multiple tasks, particularly when using Mixtral-8x7B as the base model.

On the other hand, evaluator LMs trained via weight merging show superior performance not only compared to jointly trained evaluator LMs but also single-format trained evaluator LMs, indicating *positive task transfer*. Also, while both benefit each other, merging the pairwise ranking evaluator LM weights improves direct assessment performance more significantly than the reverse.

### 6.2 Is the Effectiveness of Weight Merging due to Model Ensembling?

While we empirically find that weight merging is effective, the underlying reason remains unclear. A

natural assumption is that this effectiveness results from the ensembling effect of combining multiple models. To test this hypothesis, we conduct an ablation experiment where we train multiple evaluator LMs on different random seeds and merge them. Specifically, we merge two evaluator LMs trained on direct assessment formats (denoted as ‘Direct Assessment & Direct Assessment’) and two evaluator LMs trained on pairwise ranking formats (denoted as ‘Pairwise Ranking & Pairwise Ranking’). We use Mistral-7B-Instruct as our base model.

The results are presented in Table 6. Across multiple benchmarks, merging evaluator LMs trained on the same evaluation format does not enhance evaluation performance. Specifically, merging two evaluator LMs trained on the same evaluation format—whether direct assessment or pairwise ranking—negatively impacts performance on average for both direct assessment and pairwise ranking benchmarks. In contrast, merging two evaluator LMs, each trained on direct assessment and pairwise ranking formats, results in superior performance compared to the other settings. This indicates that the beneficial task transfer in weight merging arises from integrating different evaluation formats, not ensembling multiple models.

### 6.3 Quantifying Positive Transfer across Evaluation Formats

To explore how training on direct assessment feedback data influences pairwise ranking accuracy and vice versa, we experiment by adjusting the  $\alpha$  value during linear merging. We evaluate the average performance using all eight benchmarks in our experiments. To illustrate the average performance (colored in black), we adjust the scale by multiplying the Pearson correlations from direct assessment, which originally range from 0 to 1, by 100 before averaging them with the pairwise ranking accuracy.

The results are shown in Figure 3. For direct assessment benchmarks, evaluator LMs obtain the optimal performance when  $\alpha$  is set to 0.5. This indirectly indicates that both pairwise ranking and direct assessment feedback data contribute equally. On the other hand, for pairwise ranking benchmarks, the performance is optimal when  $\alpha$  is set to 0.3. This also implies that while both benefit each other, training on pairwise ranking improves direct assessment performance more than the reverse.

Training Method	DIRECT ASSESSMENT BENCHMARKS				PAIRWISE RANKING BENCHMARKS			
	Vicuna Ben.	MT Ben.	FLASK	Average	HHH Align.	MT Ben. H.J.	Auto-J Eval	Average
<b>Mistral-Instruct-7B</b>								
PROMPTING	0.486	0.284	0.480	0.417	67.42	63.82	60.94	64.06
DIRECT ASSESSMENT ONLY	0.537	<b>0.561</b>	<u>0.519</u>	<u>0.539</u>	73.33	56.76	64.38	64.82
PAIRWISE RANKING ONLY	-	-	-	-	<u>78.73</u>	<u>67.06</u>	72.03	72.61
JOINT TRAINING	<u>0.548</u>	0.450	0.457	0.485	<b>80.09</b>	65.49	<u>73.60</u>	<u>73.06</u>
WEIGHT MERGING	<b>0.666</b>	<u>0.548</u>	<b>0.659</b>	<b>0.624</b>	74.66	<b>70.78</b>	<b>75.07</b>	<b>73.50</b>
<b>Mixtral-Instruct-8x7B</b>								
PROMPTING	0.566	0.551	0.507	0.541	77.38	<u>71.42</u>	73.55	74.56
DIRECT ASSESSMENT ONLY	0.625	<u>0.664</u>	0.587	<u>0.625</u>	74.21	53.14	65.85	64.40
PAIRWISE RANKING ONLY	-	-	-	-	<u>84.16</u>	66.27	<u>75.66</u>	<u>75.36</u>
JOINT TRAINING	<u>0.628</u>	0.560	<u>0.596</u>	0.595	82.35	68.73	74.78	75.29
WEIGHT MERGING	<b>0.685</b>	<b>0.665</b>	<b>0.659</b>	<b>0.670</b>	<b>85.52</b>	<b>71.96</b>	<b>79.98</b>	<b>79.15</b>

Table 5: **Single-Format Training vs Joint Training vs Weight Merging** Pearson correlations between evaluator LMs trained with different methods and GPT-4-1106. Evaluator LMs trained with weight merging outperform single-format-trained and jointly-trained evaluator LMs across multiple benchmarks.

Training Data Evaluation Format	DIRECT ASSESSMENT BENCHMARKS				PAIRWISE RANKING BENCHMARKS			
	Vicuna Ben.	MT Ben.	FLASK	Average	HHH Align.	MT Ben. H.J.	Auto-J Eval	Average
NO TRAINING (PROMPTING)	0.486	0.284	0.480	0.417	67.42	63.82	60.94	64.06
DIRECT ASSESSMENT ONLY	0.537	<b>0.561</b>	<u>0.519</u>	<u>0.539</u>	73.33	56.76	64.38	64.82
PAIRWISE RANKING ONLY	-	-	-	-	<b>78.73</b>	<u>67.06</u>	72.03	<u>72.61</u>
DIRECT ASSESSMENT & DIRECT ASSESSMENT	<u>0.552</u>	0.493	0.505	0.517	73.30	55.00	63.69	64.13
PAIRWISE RANKING & PAIRWISE RANKING	-	-	-	-	<u>78.70</u>	65.20	<u>72.72</u>	72.21
DIRECT ASSESSMENT & PAIRWISE RANKING	<b>0.666</b>	<u>0.548</u>	<b>0.659</b>	<b>0.624</b>	74.66	<b>70.78</b>	<b>75.07</b>	<b>73.50</b>

Table 6: **Unifying Formats vs Ensembling** Pearson correlations with GPT-4-1106 (Vicuna Bench, MT Bench, FLASK) and agreement with human evaluators (HHH Alignment, MT Bench Human Judgment, Auto-J Eval). Merging models trained with the same evaluation formats (ensembling) underperforms merging models trained with different formats (unifying formats).

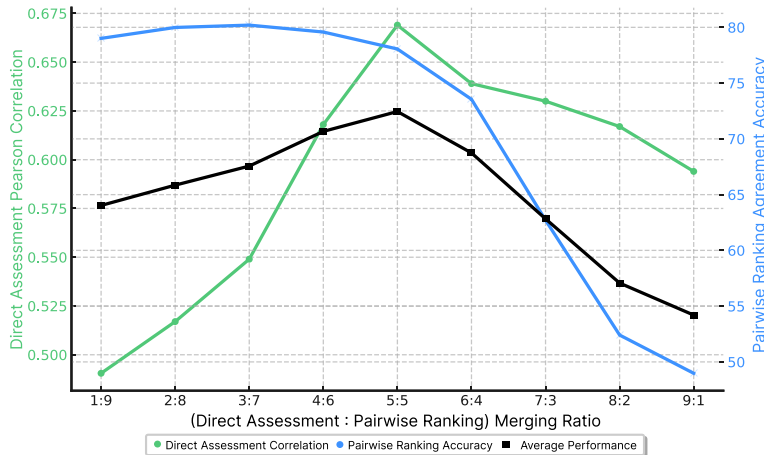


Figure 3: **When merging models, the influence of relative evaluation on absolute evaluation is greater than the influence of absolute evaluation on relative evaluation.** Performance of Direct Assessment (colored in green) and Pairwise Ranking (colored in blue) when altering the  $\alpha$  value to merge evaluator LMs trained on different formats.

## 7 Conclusion

We introduce PROMETHEUS 2, an open-source LM specialized in evaluating other responses. Unlike existing open evaluator LMs that cannot effectively process both direct assessment and pairwise rank-

ing—the two most prevalent evaluation schemes—the PROMETHEUS 2 models demonstrate superior performance on both schemes, significantly narrowing the gap with proprietary LM-based evaluations. To train the PROMETHEUS 2 models, we develop the PREFERENCE COLLECTION, the first pairwise



ranking dataset that includes over 1,000 instance-wise evaluation criteria beyond basic qualities such as helpfulness and harmlessness. Notably, we find that merging evaluator LMs trained on either direct assessment or pairwise ranking formats can lead to a unified evaluator LM with strong performance. We hope that our work encourages more research on using open-source LMs as evaluators.

## Acknowledgements

We thank the KAIST AI LKLab members for helpful discussions. This work was partly supported by LG AI Research grant (Self-improving logical reasoning capabilities of LLMs, 2024, 50%) and the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2024-00397966, Development of a Cybersecurity Specialized RAG-based sLLM Model for Suppressing Gen-AI Malfunctions and Construction of a Publicly Demonstration Platform, 50%).

## Limitations

Evaluation is fundamentally a very multi-faceted task. In this paper, we used an indirect method to assess the evaluation capability of evaluator LMs by measuring if they perform evaluations similar to human evaluators or proprietary LMs, such as GPT-4-1106 and Claude-3-Opus. However, this may not necessarily be the best approach. Future work could explore meta-evaluation pipelines that reevaluate the results of evaluator LMs or methodologies that allow humans to efficiently review evaluation results. Also note that it is crucial to use model-based evaluations in conjunction with human evaluation instead of solely relying on it.

Additionally, the degree to which evaluator LMs can generalize was based on an analysis by Kim et al. (2023), which checked for overlap between the data used to train the evaluator LMs and the data used to evaluate them. This study extended the evaluation to eight different datasets with human judgments to check the generalization capability of evaluation under various circumstances. However, this may not be sufficient. One of the major challenges in evaluating evaluator LMs is obtaining the “evaluation results” (e.g., human judgment). Automating evaluations with LMs could greatly benefit many areas of NLP research, hence the role of future work in creating feedback benchmarks that include human judgment or data for training

evaluator LMs is crucial.

One downside of the PROMETHEUS 2 is that it operates only on a 1-5 point Likert scale for absolute evaluation or a comparative evaluation style of ‘A is better & B is better’. Depending on the use case, people may need a 1-10 point absolute evaluation, a ranking method for five responses at once, or a checklist-based evaluation not covered in the paper. While proprietary LMs can flexibly conduct evaluations in any format if a well-described prompt is devised, open-source LMs cannot produce good evaluation results without training, and conversely, if trained in one or two formats, they lose the flexibility to conduct different evaluations. Future work could examine whether evaluator LMs trained in each format, as done in this paper, can handle evaluations for added formats well when weight merging is employed.

Lastly, the paper presents an evaluation model that can handle both absolute and comparative evaluation formats well through weight merging based on empirical experiments. However, fundamentally explaining why weight merging works well remains a challenging task. To address this, Section 6 indirectly analyzes the effectiveness of weight merging by comparing it with joint training, demonstrating that the improvement in evaluation performance is not due to model ensembling, and showing that the impact of comparative evaluation on absolute evaluation is greater than the reverse. Our best current interpretation is that “absolute and comparative evaluations are not completely different tasks, so weight merging could handle both without degeneration, and conversely, because they are not too similar, weight merging performed better than joint training.” Future work could theoretically analyze this or further explore whether weight merging can effectively work in fields other than LLM evaluation.

## References

- Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. [A general language assistant as a laboratory for alignment](#).
- 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*.
- Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.
- 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](#).
- Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.
- Shachar Don-Yehiya, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. 2022. Cold fusion: Collaborative descent for distributed multitask finetuning. *arXiv preprint arXiv:2212.01378*.
- Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. AlpacaFarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.
- Markus Freitag, David Grangier, and Isaac Caswell. 2020. Bleu might be guilty but references are not innocent. *arXiv preprint arXiv:2004.06063*.
- Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2024. Llm-based nlg evaluation: Current status and challenges. *arXiv preprint arXiv:2402.01383*.
- Sebastian Gehrmann, Tosin Adewumi, Karmanya Agarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh Dhole, et al. 2021. The gem benchmark: Natural language generation, its evaluation and metrics. *arXiv preprint arXiv:2102.01672*.
- Sebastian Gehrmann, Abhik Bhattacharjee, Abinaya Mahendiran, Alex Wang, Alexandros Pappangelis, Aman Madaan, Angelina McMillan-Major, Anna Shvets, Ashish Upadhyay, Bingsheng Yao, et al. 2022. Gemv2: Multilingual nlg benchmarking in a single line of code. *arXiv preprint arXiv:2206.11249*.
- Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee’s mergekit: A toolkit for merging large language models. *arXiv preprint arXiv:2403.13257*.
- Suchin Gururangan, Margaret Li, Mike Lewis, Weijia Shi, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. 2023. Scaling expert language models with unsupervised domain discovery. *arXiv preprint arXiv:2303.14177*.
- Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of bertscore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.
- Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023a. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*.
- Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2023b. Exploring the benefits of training expert language models over instruction tuning. *arXiv preprint arXiv:2302.03202*.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- 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 Wenhao Chen. 2023b. Tigerscore: Towards building explainable metric for all text generation tasks. *arXiv preprint arXiv:2310.00752*.
- Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023c. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*.
- Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoon 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, Ji Yong Cho, Shayne Longpre, Chaeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, et al. 2024. The biggen bench: A principled benchmark for fine-grained evaluation of language models with language models. *arXiv preprint arXiv:2406.05761*.

- 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*.
- Seongyun Lee, Seungone Kim, Sue Hyun Park, Geewook Kim, and Minjoon Seo. 2024. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation. *arXiv preprint arXiv:2401.06591*.
- Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023a. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*.
- Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. 2022. Branch-train-merge: Embarrassingly parallel training of expert language models. *arXiv preprint arXiv:2208.03306*.
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. AlpacaEval: An automatic evaluator of instruction-following models. [https://github.com/tatsu-lab/alpaca\\_eval](https://github.com/tatsu-lab/alpaca_eval).
- Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, and Chongyang Tao. 2024. Leveraging large language models for nlg evaluation: A survey. *arXiv preprint arXiv:2401.07103*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- 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.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. Gpt-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.
- Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716.
- 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*, pages 311–318.
- Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2024. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36.
- Natalie Schluter. 2017. The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 41–45. Association for Computational Linguistics.
- Sainbayar Sukhbaatar, Olga Golovneva, Vasu Sharma, Hu Xu, Xi Victoria Lin, Baptiste Rozière, Jacob Kahn, Daniel Li, Wen-tau Yih, Jason Weston, et al. 2024. Branch-train-mix: Mixing expert llms into a mixture-of-experts llm. *arXiv preprint arXiv:2403.07816*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrutu Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. *Llama 2: Open foundation and fine-tuned chat models*.
- Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. 2024. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*.
- Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Y Wu, and Zhifang Sui. 2023a. Math-shepherd: A label-free step-by-step verifier for llms in mathematical reasoning. *arXiv preprint arXiv:2312.08935*.
- Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. 2023b. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*.
- Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. *Ad-*

*vances in Neural Information Processing Systems*,  
36.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2023. Flask: Fine-grained language model evaluation based on alignment skill sets. *arXiv preprint arXiv:2307.10928*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2023. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *arXiv preprint arXiv:2311.03099*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.

Verification Standards	RESULTS
<b>Coherence</b>	99.5 % (Passed)
<b>Suitability</b>	98.5 % (Passed)
<b>Criticality</b>	88% (Win rate)

Table 7: Human verification results to assess the quality of the PREFERENCE COLLECTION. We use three standards to assess the quality of verbal feedback  $v_{r_m, r_n}$ .

<b>Temperature</b>	1.0
<b>Top_p</b>	0.9
<b>Max New Tokens</b>	1024
<b>Repetition Penalty</b>	1.03

Table 8: Hyperparameters used to inference different evaluator LM baselines.

<b>Base Model</b>	mistralai/Mistral-7B-Instruct-v0.2
<b>Torch dtype</b>	bfloat16
<b>Epoch</b>	1
<b>Train Data 1</b>	FEEDBACK COLLECTION
<b>Train Data 2</b>	PREFERENCE COLLECTION
<b>Max Seq Length</b>	4096
<b>Learning Rate</b>	1e-5
<b>Train Batch Size</b>	4
<b>Random Seed</b>	42
<b>Merging Strategy</b>	Linear ( $\alpha = 0.5$ )
<b>Training Method</b>	Supervised Fine-tuning

Table 9: Hyperparameters used to train PROMETHEUS 2 7B.

<b>Base Model</b>	mistralai/Mixtral-8x7B-Instruct-v0.1
<b>Torch dtype</b>	bfloat16
<b>Epoch</b>	1
<b>Train Data 1</b>	FEEDBACK COLLECTION
<b>Train Data 2</b>	PREFERENCE COLLECTION
<b>Max Seq Length</b>	4096
<b>Learning Rate</b>	1e-5
<b>Train Batch Size</b>	8
<b>PEFT</b>	True
<b>Lora_r</b>	256
<b>Lora_alpha</b>	512
<b>Lora_Dropout</b>	0.1
<b>Lora Target Module</b>	Q proj,K proj,V proj,O proj,W proj,LM_Head
<b>Random Seed</b>	42
<b>Merging Strategy</b>	DARE Merging
<b>Merging p</b>	0.1
<b>Merging Lambda</b>	1.95
<b>Training Method</b>	Supervised Fine-tuning

Table 10: Hyperparameters used to train PROMETHEUS 2 8x7B.

## A Quality Verification of the PREFERENCE COLLECTION

To ensure the quality of the PREFERENCE COLLECTION, particularly the generated verbal feedback  $v_{r_m, r_n}$ , we employ five annotators with backgrounds in natural language processing. The annotation study was designed and administered in accordance with [Affiliation X]’s ethical guidelines. Crowd workers were informed of the potential risks

of participation and researcher contact information before hand in the study consent form. The hourly wage and expected study time were informed in the Prolific platform. We compensated workers 9 GBP per hour. 3 were from USA and 2 were from Asian demographics.

We randomly sample 200 instances with different instructions and conduct a three-part verification process. First, we assess the **coherence** of  $v_{r_m, r_n}$  with the scoring decision (*i.e.*, ‘A is better’ or ‘B is better’). Second, we evaluate the **suitability** of  $v_{r_m, r_n}$  against the evaluation criteria  $e$ . Lastly, to determine the **criticality** of the feedback, we compare the newly generated  $v_{r_m, r_n}$  with a concatenation of  $v_{r_m}$  and  $v_{r_n}$ . This aims to determine if  $v_{r_m, r_n}$  effectively leverages the mutual information between  $r_m$  and  $r_n$ . Annotators then vote on whether  $v_{r_m, r_n}$  or the concatenation of  $r_m$  and  $r_n$  is more critical. The results are shown in Table 7. Note that the Preference Collection only includes English instances.

## B Training and Inference Details

The configurations we used for prompting and training evaluator LMs are shown in Table 8, 9, 10. For Auto-J, PairRM and UltraRM, we utilize their prompt template, inference hyperparameter mentioned in the model cards or github repositories in order to ensure the configuration is optimal for a fair performance comparison. For proprietary LMs, PROMETHEUS 1, and PROMETHEUS 2 models, we use the same prompt template and evaluation configurations.

For both training and inference, we utilized eight 40GB NVIDIA A100 GPUs. Training required approximately 800 GPU hours, using the implementation from the Alignment Handbook repository<sup>2</sup>. For inference, we used the vllm framework<sup>3</sup>.

The results from Direct Assessment are averaged after three multiple runs, and pairwise grading is conducted in a single run. Instead of using error bars, we report the consistency in assessment formats, Krippendorff’s alpha for consistency in direct assessment, and transitivity statistics for consistency in pairwise ranking.

## C Direct Assessment Results: Extended

Table 11 and 12 (on the next page) shows the extended results Table 3. Even when changing the

<sup>2</sup><https://github.com/huggingface/alignment-handbook>

<sup>3</sup><https://github.com/vllm-project/vllm>

metrics to either Kendall-Tau and Spearman, the overall trends are maintained. PROMETHEUS 2 shows superior evaluation performances among the open evaluator LMs, achieving high correlations with humans and proprietary LMs.

## **D License**

Our models are released under the Apache 2.0 license. The Preference Collection dataset is subject to OpenAI's Terms of Use for generated data. The model could be used for commercial purposes while the dataset is intended for research purposes. We used perspective API to ensure that the training data or evaluation datasets do not include PII-included instances.

Evaluator LM	VICUNA BENCH		MT BENCH		FLASK			Feedback Bench
	GPT-4-1106	Claude-3-Opus	GPT-4-1106	Claude-3-Opus	GPT-4-1106	Claude-3-Opus	Humans	GPT-4-0613
LLAMA2-CHAT 7B	0.183	0.203	0.065	0.070	0.229	0.186	0.211	0.419
LLAMA2-CHAT 13B	0.145	0.146	-0.019	0.037	0.160	0.174	0.174	0.453
LLAMA2-CHAT 70B	0.282	0.382	0.150	0.196	0.310	0.310	0.231	0.487
MISTRAL-INSTRUCT-7B	0.314	0.391	0.208	0.281	0.395	0.384	0.287	0.454
MIXTRAL-INSTRUCT-8X7B	0.395	0.468	<u>0.433</u>	<u>0.419</u>	0.410	0.408	0.304	0.551
PROMETHEUS-7B	0.405	0.425	0.290	0.263	0.282	0.251	0.236	0.770
PROMETHEUS-13B	0.397	0.434	0.299	0.352	0.365	0.352	0.299	<u>0.793</u>
AUTO-J (13B)	0.282	0.242	0.303	0.272	0.312	0.282	0.312	0.515
PROMETHEUS-2-7B	<u>0.543</u>	<u>0.476</u>	0.390	0.372	<u>0.476</u>	<u>0.446</u>	<u>0.377</u>	0.784
PROMETHEUS-2-8X7B	<b>0.559</b>	<b>0.515</b>	<b>0.535</b>	<b>0.483</b>	<b>0.526</b>	<b>0.507</b>	<b>0.388</b>	<b>0.800</b>
GPT-3.5-TURBO-0613	0.255	0.287	0.148	0.157	0.360	0.315	0.298	0.489
GPT-4-1106	/	0.553	/	0.590	/	0.609	0.517	0.662
CLAUDE-3-OPUS	0.553	/	0.590	/	0.609	/	0.453	0.693

Table 11: Kendall-Tau correlations between reference evaluators (listed on top) and evaluator LMs. The best comparable statistics are **bolded** and second best underlined except proprietary LMs.

Evaluator LM	VICUNA BENCH		MT BENCH		FLASK			Feedback Bench
	GPT-4-1106	Claude-3-Opus	GPT-4-1106	Claude-3-Opus	GPT-4-1106	Claude-3-Opus	Humans	GPT-4-0613
LLAMA2-CHAT 7B	0.236	0.255	0.084	0.089	0.301	0.244	0.279	0.511
LLAMA2-CHAT 13B	0.178	0.179	-0.025	0.044	0.206	0.222	0.224	0.543
LLAMA2-CHAT 70B	0.348	0.466	0.197	0.252	0.391	0.389	0.298	0.585
MISTRAL-INSTRUCT-7B	0.389	0.480	0.266	0.358	0.499	0.478	0.374	0.563
MIXTRAL-INSTRUCT-8X7B	0.476	0.556	<u>0.545</u>	<u>0.517</u>	0.505	0.500	0.386	0.659
PROMETHEUS-7B	0.508	0.528	0.385	0.349	0.367	0.326	0.317	0.876
PROMETHEUS-13B	0.492	0.534	0.401	0.470	0.474	0.454	0.398	<u>0.893</u>
AUTO-J (13B)	0.337	0.297	0.408	0.365	0.402	0.358	0.408	0.623
PROMETHEUS-2-7B	<b>0.664</b>	<u>0.591</u>	0.509	0.482	<u>0.597</u>	<u>0.555</u>	<u>0.491</u>	0.885
PROMETHEUS-2-8X7B	<u>0.660</u>	<b>0.615</b>	<b>0.669</b>	<b>0.605</b>	<b>0.642</b>	<b>0.618</b>	<b>0.496</b>	<b>0.912</b>
GPT-3.5-TURBO-0613	0.319	0.354	0.192	0.198	0.446	0.390	0.374	0.565
GPT-4-1106	/	0.659	/	0.721	/	0.729	0.650	0.753
CLAUDE-3-OPUS	0.659	/	0.721	/	0.729	/	0.567	0.784

Table 12: Spearman correlations between reference evaluators (listed on top) and evaluator LMs. The best comparable statistics are **bolded** and second best underlined except proprietary LMs.

Evaluator LM	HHH ALIGNMENT			MT BENCH HUMAN JUDG.			AUTO-J EVAL		
	Direct2Pair(↑)	Pair2Pair(↑)	$\Delta(\downarrow)$	Direct2Pair(↑)	Pair2Pair(↑)	$\Delta(\downarrow)$	Direct2Pair(↑)	Pair2Pair(↑)	$\Delta(\downarrow)$
AUTO-J (13B)	46.61	<u>75.57</u>	28.96	48.14	69.12	20.98	47.40	<u>76.64</u>	29.24
PROMETHEUS-2-7B	<u>74.21</u>	74.66	<b>0.45</b>	<b>63.24</b>	<u>70.78</u>	<b>7.54</b>	<b>68.11</b>	75.07	<b>6.96</b>
PROMETHEUS-2-8X7B	<b>81.45</b>	<b>85.52</b>	<u>4.07</u>	<u>61.67</u>	<b>71.96</b>	<u>10.29</u>	<u>66.54</u>	<b>79.98</b>	<u>13.44</u>
GPT-4-1106-PREVIEW	83.71	90.95	7.24	68.04	79.90	11.86	54.27	83.12	28.85
CLAUDE-3-OPUS	84.62	94.57	9.95	62.65	77.65	15.00	61.04	82.90	21.86

Table 13: **Consistency across Evaluation Formats** Pairwise ranking accuracy when assessing in direct assessment formats (denoted as ‘Direct2Pair’) and pairwise ranking formats (denoted as ‘Pair2Pair’). Smaller  $\Delta$  values indicate that evaluator LMs can robustly evaluate across the two different formats.

Evaluator LM	Vicuna Ben.	MT Ben.	FLASK
LLAMA2-CHAT 7B	0.3558	0.2565	0.4379
LLAMA2-CHAT 13B	0.2017	0.2998	0.4038
LLAMA2-CHAT 70B	0.5212	0.4559	0.6204
MISTRAL-INSTRUCT-7B	0.5157	0.4393	0.5884
MIXTRAL-INSTRUCT-8x7B	0.5459	0.6229	0.6976
PROMETHEUS-7B	0.6049	0.5363	0.5970
PROMETHEUS-13B	0.5734	0.5181	0.5624
AUTO-J (13B)	0.4976	0.5069	0.6160
PROMETHEUS-2-7B	0.6018	0.5340	0.5991
PROMETHEUS-2-8x7B	<b>0.6383</b>	<b>0.6862</b>	<b>0.7874</b>
GPT-3.5-TURBO-0613	0.7108	0.4800	0.6389
GPT-4-1106-PREVIEW	0.7366	0.8271	0.8355
CLAUDE-3-OPUS	0.8284	0.8601	0.8976

Table 14: Krippendorff’s alpha statistics for evaluator LMs when prompted 3 times via non-deterministic decoding.

Evaluator LM	PREFERENCE COLLECTION
	Transitivity
MISTRAL-INSTRUCT-7B	87.10
MIXTRAL-INSTRUCT-8x7B	90.45
PAIR RM	91.40
ULTRA RM	94.25
AUTO-J (13B)	89.65
PROMETHEUS-2-7B	<b>97.60</b>
PROMETHEUS-2-8x7B	<u>96.75</u>
GPT-3.5-TURBO-0613	84.35
GPT-4-1106-PREVIEW	95.70
CLAUDE-3-OPUS	96.20

Table 15: Transitivity statistics to measure consistency in pairwise ranking evaluation settings.

## E Consistency of Evaluator LMs

In addition to obtaining high correlation and accuracy, achieving high consistency is another important aspect for evaluator LMs. We first test if evaluator LMs could give consistent scoring decisions in direct assessment formats. We inferencing multiple times with non-deterministic decoding (*e.g.*, using temperature 1.0). Following the experimental design from Ye et al. (2023), we choose to inference 3 times and report the Krippendorff’s alpha value. As shown in Table 14, the results indicate that training on feedback data only slightly improves consistency. On the other hand, we find that the LMs with a large number of parameters achieve high consistency. This indicates the importance of selecting a large LM as the base model when training an evaluator LM. Notably, PROMETHEUS-2-8x7B obtains the highest correlation among open evaluator LMs.

Moreover, to evaluate consistency in pairwise ranking settings (Table 15), we measure transitivity (*i.e.*, a higher score for response B over A, and for C over B, results in a higher score for C over A). As shown in Table 15, the PROMETHEUS 2

models achieve performances on par with GPT-4, showing that they could provide robust judgments in pairwise ranking schemes.

Lastly, we conduct an experiment to test if evaluator LMs could achieve consistent scores across different evaluation formats. To do this, we use pairwise ranking benchmarks and measure the performance differences when prompted with direct assessment formats and pairwise ranking formats. Specifically, following Kim et al. (2023), to process pairwise ranking datasets in a direct assessment scheme, we evaluate each response separately and compare the scoring decisions. We mark it as correct if the evaluator LM provides a higher score for the human-chosen response over the rejected one. As shown in Table 13 (on the previous page), the results show that PROMETHEUS 2 models show lower performance differences across evaluation formats, indicating their robustness.



Evaluator LM	BIGGEN BENCH			FLASK		
	Reference-free	Reference-based	$\Delta$	Reference-free	Reference-based	$\Delta$
MISTRAL-INSTRUCT	0.305	0.310	0.005	0.331	0.374	0.043
MIXTRAL-INSTRUCT	0.320	0.322	0.002	0.377	0.386	0.009
PROMETHEUS-2-7B	0.403	0.455	0.052	0.425	0.545	0.120
PROMETHEUS-2-8x7B	0.424	0.472	0.048	0.411	0.555	0.144
GPT-3.5-TURBO-0613	0.236	0.252	0.016	0.354	0.374	0.020
GPT-4-1106	0.554	0.599	0.045	0.616	0.679	0.063

Table 16: Pearson correlations between different evaluator models with and without the reference answer and Human. Reference-based evaluations outperform reference-free evaluations across all evaluator LMs.

Merging Method	DIRECT ASSESSMENT BENCHMARKS					PAIRWISE RANKING BENCHMARKS					Average
	VICUNA BEN.	MT BEN.	FLASK (HUMAN)	Feedback Ben.	Average	HHH ALIGN.	MT BEN. H.J.	AUTO-J	Pref. Ben.	Average	
LINEAR	0.642	0.543	<u>0.544</u>	0.878	0.652	78.73	67.25	73.80	92.45	78.06	82.93
SLERP	0.648	0.532	0.536	0.879	0.649	74.66	70.2	72.33	92.60	77.44	82.67
TASK ARITHMETIC	0.518	0.497	0.482	0.831	0.582	<b>80.09</b>	<u>69.80</u>	72.82	93.00	<b>78.93</b>	81.01
TIES	0.534	<b>0.567</b>	0.529	0.826	0.614	<u>79.64</u>	67.75	72.91	<b>93.95</b>	<u>78.56</u>	80.58
DARE-TIES	<u>0.653</u>	0.545	0.543	<u>0.880</u>	<u>0.655</u>	<u>79.64</u>	66.57	<u>74.68</u>	<u>93.30</u>	78.55	<u>83.27</u>
DARE-LINEAR	<b>0.666</b>	<u>0.548</u>	<b>0.545</b>	<b>0.882</b>	<b>0.660</b>	74.66	<b>70.78</b>	<b>75.07</b>	93.25	78.44	<b>83.32</b>

Table 17: Pearson correlations and accuracy measurements across various benchmarks for different merging methods. The best comparable statistics are **bolded** and second best underlined.

## F Reference-free Evaluation in Direct Assessment Formats

In this section, we assess the impact of excluding a reference answer in evaluations conducted using direct assessment formats. The results are presented in Table 16 (on the previous page). For this experiment, we employ FLASK (Ye et al., 2023) which includes human judgments and additionally the BiGGen Bench (Kim et al., 2024). The BiGGen Bench is a generation benchmark which includes a evaluation criteria tailored to each instance and provides 2840 human judgments (excluding the multilingual tasks) in direct assessment formats.

Across both benchmarks and different evaluator LM variants, the correlation with humans diminishes when the reference answer is discarded. Even for GPT-4-1106, there is a significant performance degradation (0.045, 0.063). This suggests that including a reference answer is crucial for conducting effective evaluations with LMs. Interestingly, PROMETHEUS-2-7B achieves better performance in a reference-free setting (0.403, 0.425) than Mistral-7B-Instruct-v0.2 (0.310, 0.374). Similar trends are observed for PROMETHEUS-2-8x7B (0.424, 0.411) and Mistral-8x7B-Instruct-v0.1 (0.322, 0.386). This implies that one effect of training an evaluator LM with a reference answer included is to induce the ability to ground judgments to the given reference answer.

## G Merging Method Ablation

In this section, in addition to **linear merging**, we also test different merging techniques including:

- **Slerp merging** (Goddard et al., 2024) operates by interpolating two weights  $\theta_d$  and  $\theta_p$  while preserving the geometric properties of the spherical space in which  $\theta_d$  and  $\theta_p$  reside. Specifically, this is conducted by normalizing  $\theta_d$  and  $\theta_p$  into unit length and then merging the two weights based on the coefficient  $\alpha$  such as:

$$\theta_{final} = \alpha \times \frac{\theta_d}{\|\theta_d\|} + (1 - \alpha) \times \frac{\theta_p}{\|\theta_p\|} \quad (4)$$

- **Task Arithmetic merging** (Ilharco et al., 2022) which can be expressed as follows:

$$\theta_{final} = \theta_{init} + \alpha \times (\theta_d - \theta_{init}) + (1 - \alpha) \times (\theta_p - \theta_{init}) \quad (5)$$

where  $\theta_{init}$  is the weight of the base model. However, we empirically find that the resulting evaluator LM  $\theta_{final}$  often does not generate valid scoring decisions (e.g., generating an integer during pairwise ranking).

- **TIES merging** (Yadav et al., 2024), while similar to Task Arithmetic merging, adds (1) a `Trim` operation to remove redundant weights in the task vector  $\theta_d - \theta_{init}$  and  $\theta_p - \theta_{init}$  and (2) `Elect` and `Disjoint` operations to resolve disagreement (i.e., opposite directed weights) between  $\theta_d - \theta_{init}$  and  $\theta_p - \theta_{init}$ .
- **DARE merging** (Yu et al., 2023), while also similar to Task Arithmetic and TIES merging, performs a `Random Drop` and `Re-scale` operations in the task vector  $\theta_d - \theta_{init}$  and  $\theta_p - \theta_{init}$  to remove redundant weights. We find that DARE merging work best when we choose Mistral-8x7B as our base model. **DARE-linear merging** is what was originally proposed by Yu et al. (2023). In **DARE-TIES merging**, the `Elect` operation from Yadav et al. (2024) is additionally added after the `Re-scale` operation.

We conduct our experiments based on the implementation from MergeKit (Goddard et al., 2024).<sup>4</sup>

In Table 17 (on the previous page), we measure the performance of evaluator LMs employing different merging methods. In direct assessment benchmarks, DARE-Linear achieves the best performance, followed by DARE-TIES and Linear merging. In pairwise ranking benchmarks, Task Arithmetics achieves the best performance, with only a minimal difference compared to other methods. On average, DARE-Linear performs best. Based on these results, we have trained Prometheus-2-7B with DARE-Linear merging. We also opted to train Prometheus-2-8x7B using DARE-Linear merging. Although the optimal merging method might differ, we have not conducted additional experiments due to computational limitations. Future work could explore whether these findings hold true.

<sup>4</sup><https://github.com/arcee-ai/mergekit>

## H PREFERENCE COLLECTION Augmentation Prompt

### Prompt for Generating Verbal Feedback in Pairwise Ranking

###Task Description:

An instruction (might include an Input inside it), two responses to evaluate (denoted as Response A and Response B), a reference answer, and a score rubric representing a evaluation criteria are given.

1. Write a detailed feedback explaining why {sub\_str}, focusing strictly on the aspects highlighted in the evaluation criteria.
2. While writing the feedback, make comparisons between Response A, Response B, and Reference Answer. Instead of examining Response A and Response B separately, go straight to the point and mention about the commonalities and differences between them.
3. While writing the feedback, do not start by mentioning {sub\_str} in the first sentence. Instead, try to write a reasoning process that delves into the commonalities and differences of the two responses and mention {sub\_str} at the last part of your justification.
4. Within the feedback, do not explicitly mention about the reference answer. For instance, do not use phrases like "Compared to the reference answer". Assume that you inherently know the reference answer which could be used to determine details that are not present in both responses under assessment.
5. Please do not generate any other opening, closing, and explanations. Just write the feedback.
6. Within the feedback, generate a string phrase "[END]" after you are finished.

###Instruction: {instruction}

###Response A: {response\_A}

###Response B: {response\_B}

###Reference Answer: {reference\_answer}

###Score Rubric: {criteria}

###Feedback:

## I Direct Assessment Prompt

### Direct Assessment System Prompt

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

### Direct Assessment Prompt Template

###Task Description:

An instruction (might include an Input inside it), a response to evaluate, and a score rubric representing a evaluation criteria are given.

1. Write a detailed feedback that assess the quality of the response strictly based on the given score rubric, not evaluating in general.
2. After writing a feedback, write a score that is an integer between 1 and 5. You should refer to the score rubric.
3. The output format should look as follows: "Feedback: (write a feedback for criteria) [RESULT] (an integer number between 1 and 5)"
4. Please do not generate any other opening, closing, and explanations.

###The instruction to evaluate:

{orig\_instruction}

###Response to evaluate:

{orig\_response}

###Score Rubrics:

{score\_rubric}

###Feedback:

## J Pairwise Ranking Prompt

### Pairwise Ranking System Prompt

You are a fair judge assistant assigned to deliver insightful feedback that compares individual performances, highlighting how each stands relative to others within the same cohort.

## Pairwise Ranking Prompt Template

### ###Task Description:

An instruction (might include an Input inside it), a response to evaluate, and a score rubric representing a evaluation criteria are given.

1. Write a detailed feedback that assess the quality of two responses strictly based on the given score rubric, not evaluating in general.
2. After writing a feedback, choose a better response between Response A and Response B. You should refer to the score rubric.
3. The output format should look as follows:  
"Feedback: (write a feedback for criteria)  
[RESULT] (A or B)"
4. Please do not generate any other opening, closing, and explanations.

### ###Instruction:

{orig\_instruction}

### ###Response A:

{response\_A}

### ###Response B:

{response\_B}

### ###Score Rubric:

{score\_rubric}

### ###Feedback: