LitBench: A Benchmark and Dataset for Reliable Evaluation of Creative Writing

Daniel Fein* **Sebastian Russo*** **Violet Xiang*** **Kabir Jolly**
Rafael Rafailov **Nick Haber**
Stanford University

Abstract

Evaluating creative writing generated by large language models (LLMs) remains challenging because open-ended narratives lack ground truths. Without performant automated evaluation methods, off-the-shelf (OTS) language models are employed as zero-shot judges, yet their reliability is unclear in this context. To address this gap, we introduce LitBench, a large-scale benchmark for creative writing evaluation, featuring a training corpus of 43,827 story pairs and a 2,480-pair test set curated from Reddit. Using LitBench, we benchmark existing LLM judges and train specialized reward models. Our analysis reveals that the strongest OTS judge, Claude-3.7-Sonnet, achieves only 73% agreement with human preferences. In contrast, our trained Bradley-Terry and generative reward models both reach 78% accuracy, outperforming all OTS judges. An online human study further validates our models, showing their rankings of newly generated stories align more closely with human preferences. Our work provides a large-scale reliable benchmark and specialized reward models for creative writing, establishing a crucial foundation for the future development of more capable verifiers.

1 Introduction

Rapid progress in fields like math and code generation has been driven by reinforcement learning with verifiable rewards (RLVR) [Guo et al., 2025, Team et al., 2025, Yang et al., 2025], benefiting from large-scale, objective datasets [Hendrycks et al., 2021, Gao et al., 2024, Jimenez et al., 2023, Pan et al., 2024]. In contrast, creative writing is inherently divergent; a single prompt can inspire countless valid stories, making objective verification impossible. This evaluation bottleneck has hindered progress in generative AI for storytelling. While human experts provide reliable judgments, doing so at the scale of AI-generated text is prohibitively expensive [Chakrabarty et al., 2024]. To fill this void, off-the-shelf (OTS) large language models (LLMs) are often used as zero-shot judges for tasks like summarization and dialogue [Zheng et al., 2023, Son et al., 2024]. However, their reliability for creative writing is questionable. LLM judges are known to exhibit biases toward length and style over substance, and they often lack internal consistency [Wang et al., 2023a, Wei et al., 2025, Feuer et al., 2025]. Such flaws are particularly concerning in a domain where the interplay between form and content is paramount.

To address these challenges, we introduce LitBench, a large-scale benchmark for evaluating creative writing. LitBench provides a standardized framework to (1) rigorously benchmark existing LLM judges and (2) train specialized reward models that better capture the nuances of human literary preference. It comprises a 2,480-pair test set and a 43,827-pair training corpus curated from high-quality stories on Reddit's `r/WritingPrompts`. Our evaluation reveals that even the strongest OTS judge, Claude-3.7-Sonnet, achieves only 73% agreement with human preferences. In contrast, our Bradley-Terry and Generative Reward Models (GenRMs), trained on the LitBench corpus, reach 78% accuracy, outperforming all zero-shot judges. Our investigation also yielded a counterintuitive finding: while GenRMs perform well, augmenting them with distilled chain-of-thought (CoT) rationales degrades, rather than improves, performance in this subjective domain. By providing a robust benchmark and validated reward models, our work lays the necessary groundwork for future advancements in AI-powered storytelling.

Our contributions are as follows:

- **LitBench**, a new public benchmark with a 2.5k-pair test set and a 43k-pair training corpus with rationales, to standardize creative writing evaluation.

*Equal contribution.
Correspondence: `drfein@stanford.edu`.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7740–7755

- A comprehensive benchmark of OTS LLM judges and trained reward models, showing that our custom-trained models (78% acc.) outperform the best OTS judge (73% acc.) at a fraction of the size.
- A human evaluation study validating the alignment of our trained reward models with human preferences on novel stories.

2 Related Work

2.1 Verifiable vs. Non-verifiable Domains.

Progress in automated evaluation has followed two distinct paths, dictated by the nature of the domain. In verifiable domains like math and coding, the existence of ground truth has enabled direct verification or training of robust verifiers [Zhang et al., 2024, Mahan et al., 2024, Lightman et al., 2023, Wang et al., 2023b, Kumar et al., 2024]. In contrast, non-verifiable domains rely on modeling subjective human preferences, as reinforcement Learning from Human Feedback (RLHF) has become the dominant paradigm [Ouyang et al., 2022, Stiennon et al., 2020]. To reduce the high cost of human annotation, many methods now substitute humans with LLMs as judges [Bai et al., 2022]. While this approach shows reasonable success on tasks like summarization [Zheng et al., 2023, Liu et al., 2023], the reliability of LLM judges can be compromised by biases towards stylistic features and length, raising serious concerns about their suitability for nuanced creative tasks [Feuer et al., 2025].

2.2 Evaluation of Creative Writing.

Automated evaluation of creative writing presents unique challenges due to its inherent subjectivity and lack of ground truth. As Chung et al. [2025] note, learning a robust reward model in this domain is difficult precisely because creative quality admits many valid realizations. Consequently, much prior work has relied on reference-based metrics such as BLEU or ROUGE, which are poorly suited to open-ended generation where no single correct answer exists [Fan et al., 2018a, Li et al., 2025]. Other approaches focus on narrow, measurable aspects of quality—such as coherence, diversity, or structural relations—which capture only fragments of what readers perceive as creative merit [Alihosseini et al., 2019, Li et al., 2020].

Despite this subjectivity, human judgment of creative products is not arbitrary. Classic findings in psychology show that expert readers achieve high inter-rater agreement when ranking creative prose and poetry, suggesting the existence of a learnable, if implicit, quality signal [Amabile, 1982]. Recent benchmarks have operationalized this signal in different ways. WritingBench [Wu et al., 2025] evaluates generative writing across a broad set of professional and functional domains. EQ-Bench Creative Writing v3 [Paech, 2024] provides an evaluation suite based on templated prompts and reference- or rubric-driven scoring. Both of these are designed primarily as generation benchmarks and assume the reliability of zero-shot LLM judges.

Other lines of work instead explore expert-driven or edit-based supervision of AI-generated creative writing. Chakrabarty et al. [2025b] study human-AI collaborative writing by collecting localized expert edits, yielding signals well-suited to stylistic revision but not story-level preference judgments. More recently, Chakrabarty et al. [2025a] combine the above edit-based dataset with a variety of mostly LLM-generated texts and edits labeled by experts.

In contrast to these approaches, LitBench targets *story-level* verification over *human-written* creative narratives. By adopting the methodology of the Stanford Human Preferences (SHP) dataset [Ethayarajh et al., 2022a], LitBench leverages large-scale crowd-sourced signals—here to produce an evaluation along with a training-scale dataset.

3 LitBench

LitBench is a benchmark for reward models that can judge creative writing quality, coupled with a larger preference dataset on which verifiers can be trained. Since reward hacking is a common issue when using preference-based reward models to improve model capabilities, we carefully construct both our evaluation and training set to ensure quality. We describe the procedure in detail, and demonstrate our curation procedure is indeed helpful.

3.1 Data Collection

We collect writing samples from the `r/WritingPrompts` subreddit. In this popular subreddit, users write stories in response to writing prompts, and engage with posted stories through upvotes or comments. In total, `r/WritingPrompts` has amassed over one million stories. Such a large corpus enables highly selective data filtration, leaving data for which we

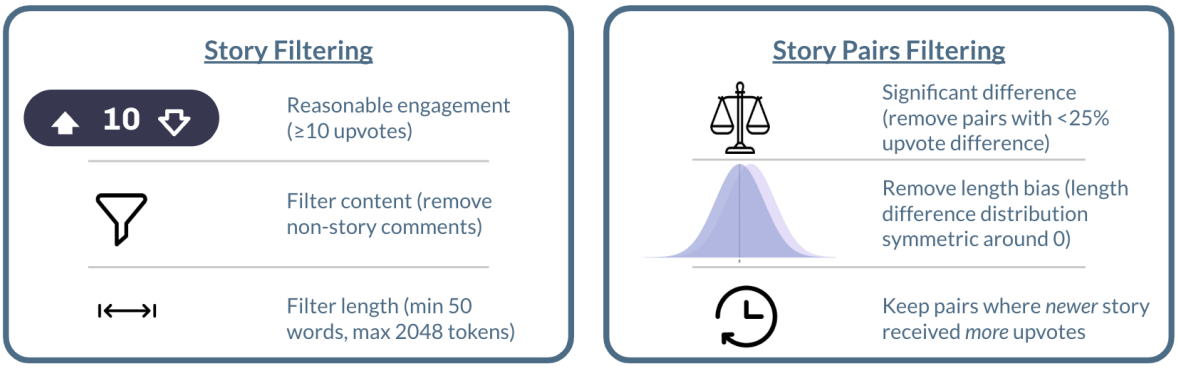

Figure 1: Preprocessing methodology for dataset creation.

can confidently assume human preferences signal. To collect the data for our benchmark, we use the Reddit API via the `praw` library. Specifically, we use the search function to collect the 100 top search results for each post collected by [Fan et al., 2018b]. This yields 5,000+ post-ids (individual prompts within the framework of the subreddit). We then construct our test set by filtering out stories older than 2023, as this data potentially overlaps with our training dataset, and is more likely to have been included in the pretraining of the models we study here. To collect our training dataset, we curate examples from the MIT-licensed `euclaise/WritingPrompts_preferences` dataset from Hugging Face*, which contains stories posted prior to 2023.

3.2 Quality Control

We begin constructing our dataset by filtering stories independently. First, to reduce the effect of noise from small upvote counts, we guarantee that each story has a reasonable amount of engagement by filtering out stories with fewer than 10 upvotes. Then, consistent with [Chung et al., 2025], we filter out stories with greater than 2048 tokens to remove excessively long stories. Lastly, we remove all entries with fewer than 50 words, because we find qualitatively that these are not sufficiently long to reflect the genre of creative fiction.

To form pairs, we carry out two steps to ensure that true preferences are being captured, and then one step to address length bias. Initially, we exclude pairs with marginal differences in upvotes, filtering out those with an upvote difference less than 25%. Next, following the methodology of [Ethayarajh et al., 2022b], we only create pairs where the higher-upvote story is also published later, mitigating temporal bias from varying exposure durations.

Lastly, we find that the resultant dataset has a length bias, with 65.25% of chosen responses longer than the rejected responses. To address this while preserving length diversity, we construct a histogram of length differences with 100 buckets and prune pairs until achieving symmetry with balanced proportions where chosen stories are both shorter and longer. This step is represented in Figure 2. The data preparation workflow is summarized in Figure 1.This entire process is performed independently for both our benchmark and training dataset.

3.3 Final Dataset Description

LitBench consists of 2,480 pairwise comparisons composed of 3,543 total stories. These stories have an average length of 550 words, and story length is right-skewed, with a tail of longer stories. We find that many of our rejected stories have an upvote-count near our prescribed minimum of 10 upvotes, with a long tail of higher rated stories. Our chosen stories have a minimum upvote count of 14, due to our decision to prune pairwise examples with an upvote differential of less than 25% of the chosen response. These distributions are shown in Figure 3.

The training dataset consists of 50,309 unique stories that are used in 43,827 pairwise-examples. The distribution is similar to the test set with respect to story lengths and upvote distribution, but the stories in the training set come strictly before 2023. The vast majority of stories were posted between 2014 and 2022. We confirm the quality of annotated training set is indeed higher than the original set by comparing reward models trained

*WritingPromptsHuggingface

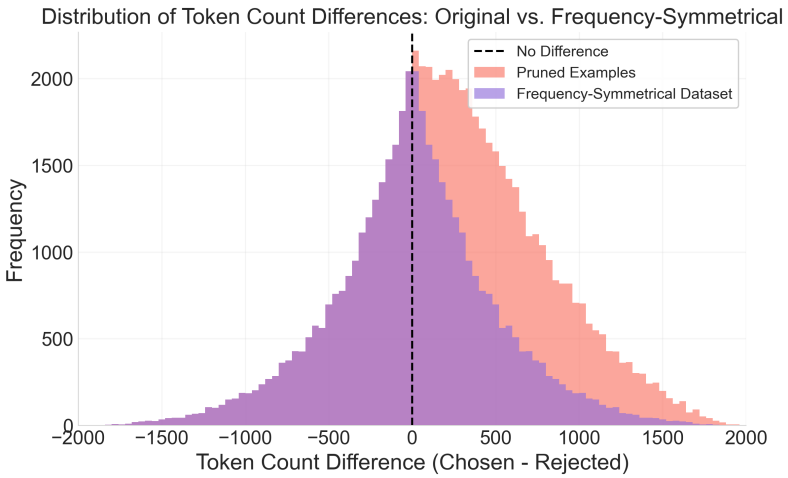

Figure 2: Length bias mitigation.

on them in Section 5.

3.4 What Does LitBench Encode?

3.4.1 Interpretability-Based Analysis

Reddit upvotes are an imperfect proxy for creative quality as they may also reflect community norms or other contextual factors. To study whether our filtration process resulted in pairwise preferences reflecting consistent writing-quality signals, we apply the interpretable preference-decomposition method of Movva et al. [2025], which uses sparse autoencoder (SAE) features to identify human-interpretable factors predictive of pairwise preferences.

Concretely, we train a small SAE on embeddings of pairwise examples from LitBench, and then train a simple linear preference model on top of these interpretable features. This lightweight model achieves **61.2%** accuracy on held-out LitBench pairs, indicating substantial preference structure recoverable from sparse features alone.

Interpreting the features associated with the highest-magnitude coefficients reveals factors that align with recognizable literary attributes of including introspection, emotional resonance, meta-narrative structure, and tonal shifts. The predictiveness of these features, rather than metadata-like popularity artifacts alone, indicates that LitBench preferences encapsulate aspects of writing quality. The full list of interpreted high-coefficient features and experimental details are available in Appendix A.1.

3.4.2 Qualitative Analysis

To examine the character of "winning" samples systematically, we read and annotated 50 pairwise writing samples from LitBench. We present a few observations below.

Why do stories win? The preferred stories often contain an unexpected twist or surprising humor; we observed many clever punchlines and wordplay. For example, we read about a tyrant queen who won over her opposition not by warfare but absurdist politeness, subverting reader expectations. Another told the story of a woman and her powerful captor named "Decimator".

Why do stories lose? Although some stories were difficult to distinguish, many felt dry and lacked emotional qualities. We found some stories were challenging to finish, due to confusing narratives or strange diction, including introducing too many characters in a short story. Of note, grammatical errors and narrative incoherence, while present in occasional losing samples, *do not* generally characterize them. We give two short examples from the dataset in Table 1.

4 Training and Evaluation Protocols

We evaluate various approaches to verification including zero-shot Bradley-Terry discriminative reward models, and generative reward models with and without chain-of-thought generation.

4.1 Bradley-Terry Discriminative Reward Models

We train a discriminative reward model using the Bradley–Terry (BT) formulation [Bradley and Terry, 1952], where each writing sample in a pair

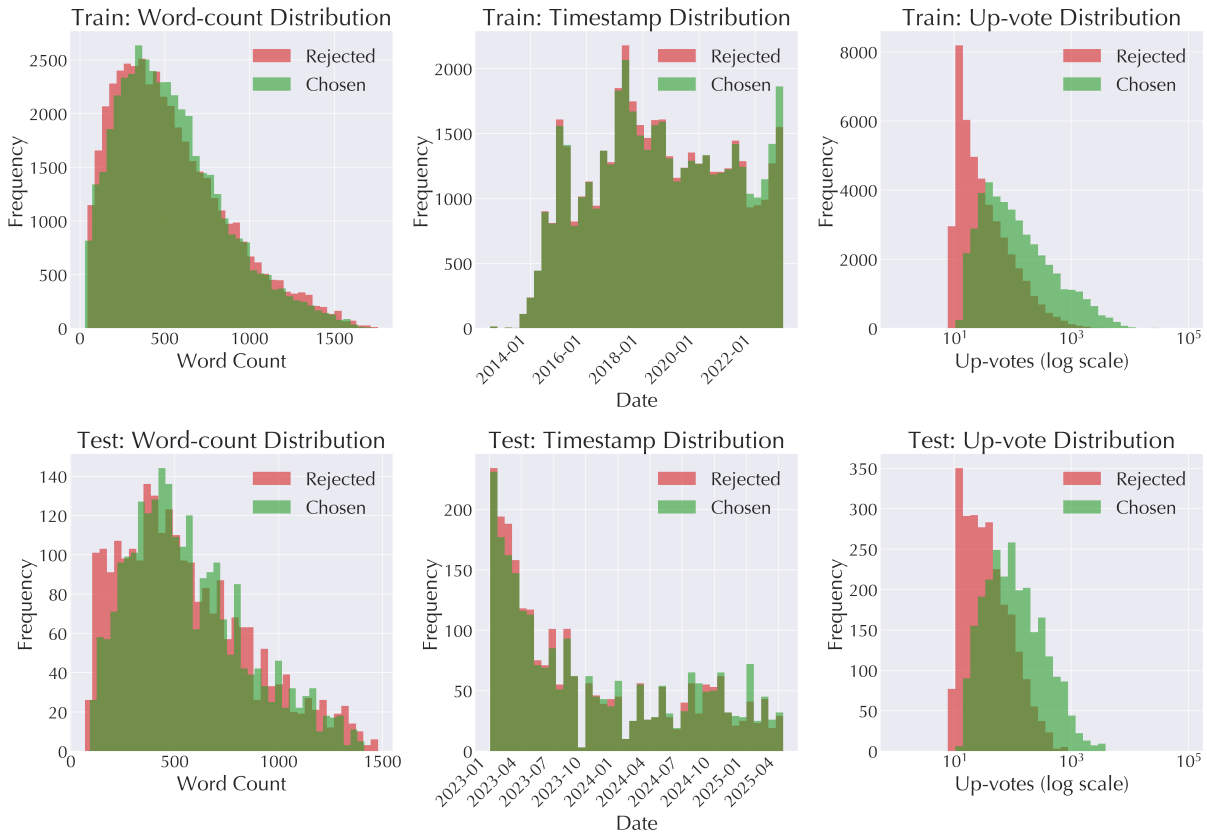

Figure 3: Distributions of word count, date, and upvotes for the LitBench test- and train-set.

is scored independently, and the model is trained to assign higher reward to the preferred sample. Given reward scores r_{chosen} and r_{rejected}, the loss is defined as:

$$\mathcal{L}_{\text{BT}} = -\log \sigma(r_{\text{chosen}} - r_{\text{rejected}}),$$

encouraging separation between better and worse samples. We append a linear layer to the base-model's last hidden state and then finetune all weights of the combined regression model. Accuracy is calculated as the percentage of cases where $r_{\text{chosen}} > r_{\text{rejected}}$.

4.2 Generative Reward Models

Generative reward models have been shown to perform well in math and coding domains, particularly for out of distribution data [Mahan et al., 2024, Zhang et al., 2024]. Generative verifiers treat classification as autoregressive generation by conducting supervised finetuning with cross-entropy loss on the predictions of an instruction-tuned model. Chain-of-thought (CoT) can also be incorporated into this process by finetuning chains of thought that precede and describe the prediction that follows. Here, we train two versions of generative reward models (GenRM): (1) GenRM - to predict which single token between "A" and "B" represents the chosen story, (2) GenRM-CoT - to reason before selecting a preferred story distilled using GPT4.1 generated rationales. We randomly shuffle the chosen and rejected stories between option A and B to avoid position bias. For both generative RMs, verdicts were collected at `temperature=0` with one sample.

4.3 Zero-shot LLM Judges

Off-the-shelf, LLM judges are shown unlabeled stories A and B, and prompted to indicate their quality preference (e.g. "A" or "B") between the pairwise samples. In particular, we instructed judges to form *explanations* prior to verdict generation. To account for the known position bias in LLM judges [Ye et al., 2024], we take the average performance of two sets of pairs, permuting the position of the stories. We selected the LLM-as-judge template prompt, specifying evaluation criteria and output format, by selecting the template with the highest precision in a validation set sampled from the training set among five hand-constructed prompts. We chose not to use automatic prompt optimiza-

Table 1: Examples of prompts with "chosen" and "rejected" completions from LitBench.

Prompt	Chosen	Rejected
In 50 words or fewer, write a story with a twist ending.	She took a deep breath and pulled back the velvet curtains. She closes her eyes and sings; her powerful voice rang loud and clear, echoing in the hall. With thunderous applause, she bows. With a sigh, she opens the microphone and pours the shampoo and begins to wash her hair.	God and Satan had many fights between before. For what reason, sometimes even they don't know. These fights often lasted days, months, even years, but this time Satan won the fight permanently. And he smeared God's blood all over the walls of the white room in the asylum.
Make me cry in two sentences.	As he laid in bed, waiting for sleep, he wished that he would wake the next morning to find that it had all been a dream, that she was alive and he wasn't alone. Don't be stupid, he thought, the only way I'll ever see her again is if I don't wake up at all.	I felt arms slide around my waist and a scent I hadn't smelled in years flooded my senses as I turned around to see my first love, the one I'd tried to, but had never been able to forget. "I told you someday I'd find my way back to you."

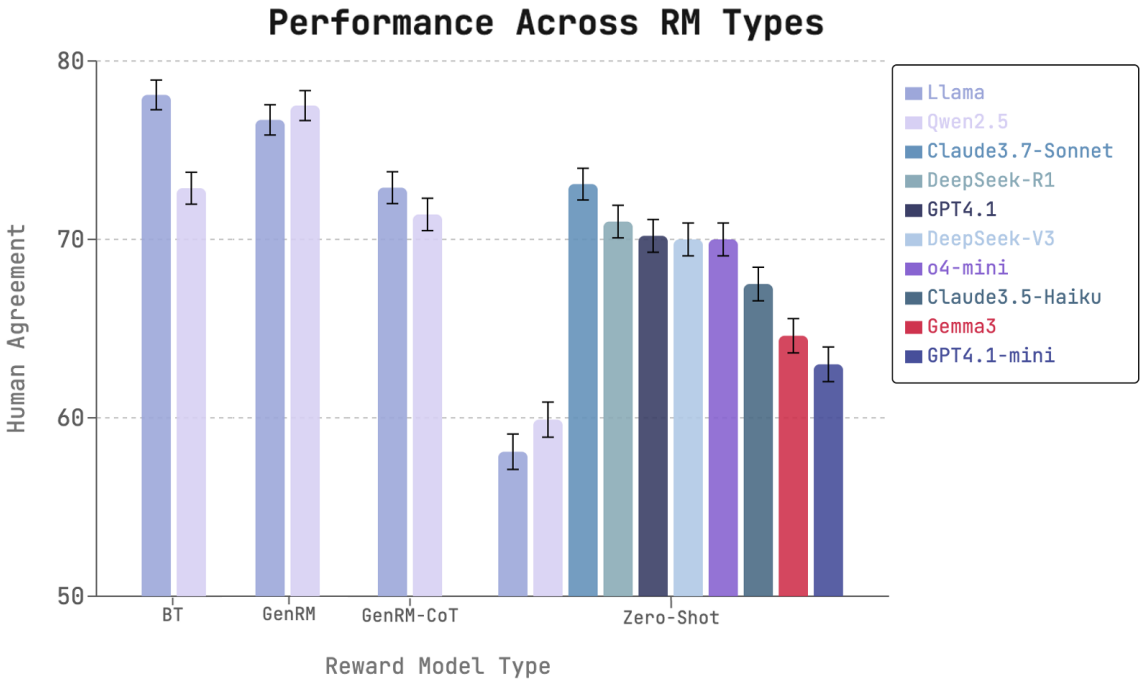

Figure 4: Trained verifiers outperform zero-shot LLM-judges on LitBench. Claude3.7-Sonnet is the strongest zero-shot model. BT verifiers are competitive with GenRMs, but GenRMs with CoTs perform worse. The sizes of Qwen, Llama and Gemma backbones are 7B, 8B and 12B, respectively.

tion tools, such as TextGrad [Yuksekgonul et al., 2025], as we observed that these methods resulted in poorer prompt performance. For further discussion on prompt optimization, prompt templates and response structure, see section A. We apply the judge methodology to a selection of state-of-the-art proprietary and open-source LLMs, and these results are demonstrated as baselines in Figure 5 and 4.

5 Results and analysis

LitBench is designed to measure how effectively learned reward models can verify quality in open-ended, creative domains. We demonstrate its utility by:

- *Validating* the construction of the dataset by benchmarking trained reward models, and evidencing reward model generality with online studies.
- *Characterizing* LLM-based methods to verify human writing, by comparing cross-model performance and analyzing their reasoning

text.

5.1 BT and Generative Reward Models Outperform Zero-Shot LLMs

We offer comparative reward model performance in Figure 4. The best Bradley–Terry reward model (Llama-8B) finetuned on LitBench training set achieves 78% human agreement, marginally surpassing the best generative verifier (GenRM-Qwen). Both GenRM and BT reward models significantly outperform the strongest zero-shot judge (Claude-3.7-Sonnet, 73%). Interestingly, adding chain-of-thought reasoning to GenRMs lowers accuracy to 72%, indicating that explicit sequential reasoning, while beneficial in math and coding tasks, introduces textual noise when judging narrative quality. Zero-shot performance scales unpredictably with backbone size; tested OpenAI, Anthropic and Deepseek models sit in the 70% range, while smaller open-source models hover near chance-plus (56–60%). These results underscore that targeted preference finetuning dominates parameter count for creative-writing evaluation, and that discriminative objectives remain the most reliable choice in this domain.

5.2 Reasoning Degrades Verdict Accuracy

Despite the general success of CoT-based in verifiable domains, here CoTs actually degraded reward model performance. To examine this, we computed statistics on explanation text produced by judge models, and correlated these features with verdict accuracy. We present characteristics, inspired by creative writing pedagogy [Sellers, 2021], most predictive of verdict accuracy in Figure 8. Among all models, discussions of the *plot* are most predictive of correctness (though particularly for Anthropic models) correlated with a +14.8% higher correctness among all models. However, most of the explanation text features had minimal relation with subsequent verdict accuracy. This analysis can be found in full in Section A.6.

5.3 Performance Scales Differentially by Reward Model

Figure 5 shows the performance improves at different rates as model size increases across all types of reward models. GenRMs with CoT start at lower performance at lower model sizes for both Llama and Qwen backbone, but steadily improve to 74%. However, GenRM without CoT has no significant improvement across model sizes, suggesting a much smaller model (1B or 1.5B) can be used to obtain similar performance. The performance of Bradley-Terry models has a notable performance difference due to the varied backbone, particularly in smaller models (1B/1.5B and 3B). For zero-shot judges, we observe similar effect that performance improves meaningfully as the size increase.

5.4 Validating Data Filtration Methodology

We further confirm our curation process by training ablative BT reward models on datasets produced by different filtering strategies and evaluating on the debiased LitBench test-set. We create a lightly filtered version of the original training set that only removes pairs containing stories that have less than 10 upvotes and pairs based on upvote difference, resulting in 395k pairs. We also create an unfiltered version paired by timestamp and upvote difference, resulting in 1.03M pairs. We train BT reward models with a Llama-3.2-1B backbone on these datasets. Despite having significantly more examples, we find that performance on LitBench without pairing by timestamp saturates at much lower levels (65%). Without our length filtering, we find saturation at 70%, but we also find that the the reward model is length-biased, strongly preferring the longer of the two stories in most cases. These results of this experiment are shown in Figure 6.

5.5 Human Experiments

We generate 64 stories each seeded from 40 LitBench prompts using GPT 4.1 and GPT 4o, and then rank them with our Llama-8B-based Bradley-Terry reward model. In an online human studies with 46 U.S./U.K. crowd-workers (10-13 annotators per pair), we evaluate human agreement with the RM-determined best and worst stories for each prompt. Figure 7 indicates that annotators selected the RM-preferred story 56.4% of the time versus 43.6

6 Discussion and Limitations

Our findings indicate that top-performing proprietary LLM judges are approaching the accuracy of specialized reward models, making them a viable alternative when domain-specific training data is unavailable. When training data is used, GenRMs offer strong performance, though our results surprisingly show that augmenting them with chain-of-thought rationales can be detrimental. Crucially,

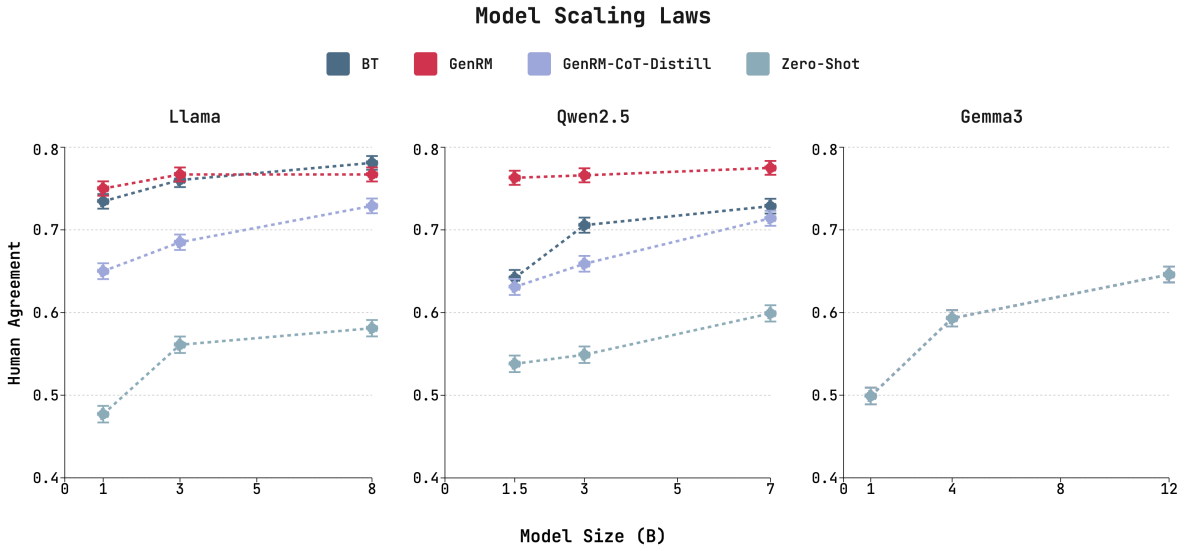

Figure 5: Human agreement scaling is inconsistent with model size for different types of RMs.

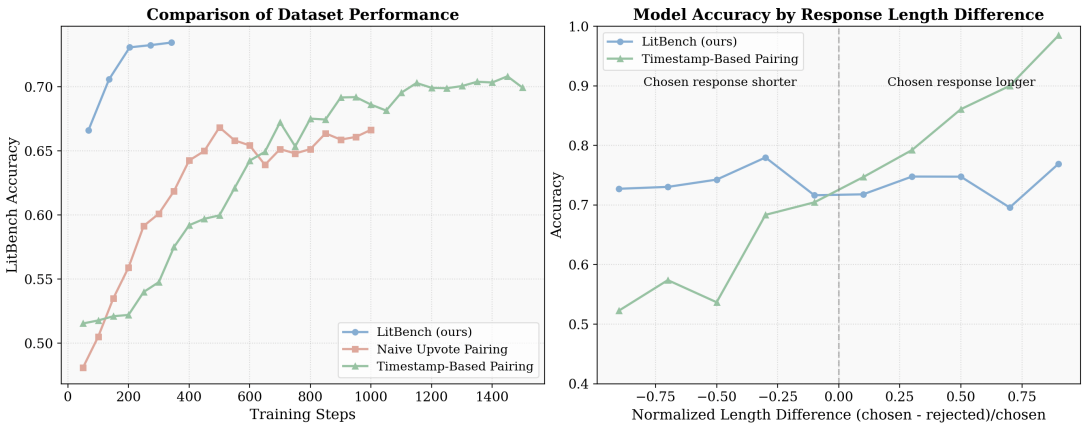

Figure 6: Naive upvote pairing and naive timestamp and upvote pairing saturate at lower accuracy than the LitBench training set. Naive timestamp and upvote pairing alone produces a length biased verifier. All models are BT RMs finetuned on a Llama-1B backbone.

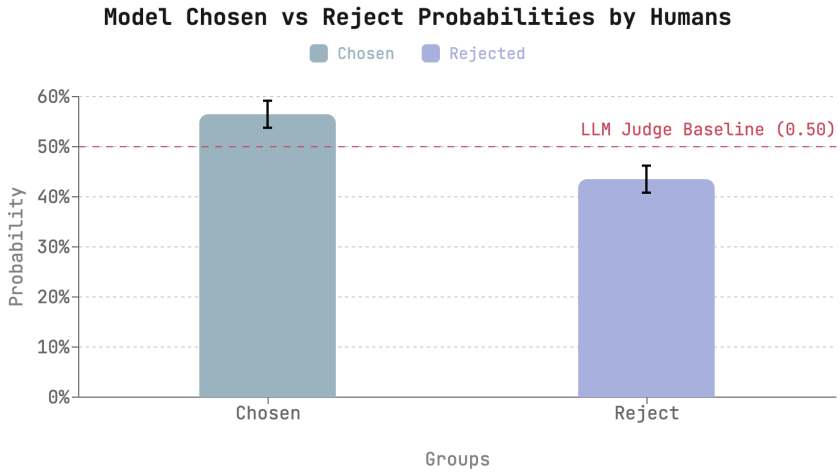

Figure 7: Human preference alignment with reward model on generated writing pairs.

the generalization of our LitBench-trained models to novel LLM-generated stories, confirmed by human evaluation, paves the way for using these verifiers to guide and improve future creative writing agents. Our work's primary limitation stems from its data source. We use Reddit upvotes as a proxy for writing quality, but this signal is inherently noisy. Upvotes can be influenced by factors beyond a story's intrinsic merit, such as social dynamics, humor, the time of posting, or an author's existing reputation. By aggregating these preferences, our benchmark simplifies the multifaceted nature of "creative quality" into a single axis of comparison. Future work could improve upon this by sourcing preference data from expert critiques or platforms with more structured feedback to build more reliable verifiers.

A Appendix

A.1 Interpretability-Based Dataset Analysis

Autointerpretation Prompt. The following prompt is used by `interpret_litbench_sae.py` to generate natural-language descriptions for learned SAE features. It extends the standard `interpret-neuron-binary` template with LitBench-specific instructions.

Autointerp (Neuron Interpretation) Prompt

```
You are a machine learning researcher who
has trained a neural network on a text
dataset.
You are trying to understand what text
features cause a specific neuron in the
neural network to fire.

You are given two sets of SAMPLES:
POSITIVE SAMPLES that strongly activate
the neuron, and
NEGATIVE SAMPLES from the same
distribution that do not activate the
neuron.

Each example corresponds to a pair of LLM
stories.
Positive activations indicate a concept
that appears more in the chosen_story,
while negative activations indicate the
concept is stronger in the rejected_story.

Describe concrete differences between the
two stories (tone, themes, writing style,
etc.).

POSITIVE SAMPLES:
----------------
{positive_texts}
----------------

NEGATIVE SAMPLES:
----------------
{negative_texts}
----------------

Rules about the feature you identify:
- The feature should be objective and
concrete.
- The feature must be present in the
positive samples and absent in the
negative samples.
- Do not describe generic features that
also appear in negative samples.
- The feature should be as specific as
possible while applying to all positive
samples.

Output exactly one feature, starting with
"-" and surrounded by quotes.
Your response is:
- "
```

Embedding model. We use `text-embedding-3-small` (OpenAI) with dimensionality 1536.

SAE architecture. We train a Matryoshka BatchTopK Sparse Autoencoder over difference embeddings (`chosen` $-$ `rejected`) to support signed features. The SAE uses $M = 32$ neurons with sparsity $K = 4$, and Matryoshka prefixes $\{8, 32\}$ to encourage hierarchical feature structure.

Training. We optimize with Adam (learning rate 5×10^{-4}, batch size 512) for up to 100 epochs with early stopping. Batch Top-K sparsity selects the top $K \times$ batch_size activations per batch. Dead neurons are revived using Aux-K activation after 256 non-firing steps.

Results. Full interpretations of highly activating features can be found in Table 2.

A.2 LLM-as-judge Raw Results

Table 3 shows LLM-as-a-judge results.

A.3 LLM-as-judge Prompt Optimization

Motivation. We sought to give out-of-the-box LLM evaluators the best chance of performing accurately on this benchmark. LLM-based judges have been demonstrated to be extremely sensitive to prompt content [Zheng et al., 2023]. In this experimental setting, prompts enable (a) introduction of criteria for judges to utilize in inferring verdicts, and (b) specification of an output format to ensure easily parsed results.

Strategy. There are numerous prompt-optimization libraries that automatically 'differentiate' the prompt text to improve accuracy on a given evaluation metric. However, after some experimentation with these, we opted to apply methods of our own design to optimize the prompts, following the approach below.

Goal: Select an optimized prompt for each family of models (e.g. Llama).

Optimization Method.

1. Hand-construct six 'template' prompts, each introducing different criteria for the judge to use when generating a verdict.
2. Standardize output format: request JSON objects from large instruction-tuned models; request plaintext from smaller models.
3. Using a midrange model for each family, evaluate each prompt with a validation set ($n = 500$) drawn from the training set to avoid bias.
4. Adopt the prompt, by family, that yields the

Interpreted Feature	Coef.	Prevalence
Narrative includes detailed internal thoughts and emotional self-reflection by the protagonist	0.178	0.270
Meta-narrative or self-referential storytelling (e.g., breaking the fourth wall)	0.157	0.109
Dialogue-driven scenes featuring wit, irony, or dark humor	0.139	0.121
Emotionally warm depictions of close family relationships	0.107	0.264
Reflection on personal impact of unusual or reality-altering events	0.086	0.094
First-person narration with sustained focus on the narrator's own perspective	−0.288	0.274
Extended world-building or lore exposition embedded in the narrative	−0.084	0.107
Dialogue-heavy exposition with limited narrative grounding	−0.074	0.113
Abrupt tonal shift from despair to hope within a short span	−0.073	0.051
Vivid depiction of pain, illness, or death with sensory emphasis	−0.054	0.043

Table 2: Representative interpretable SAE features predictive of LitBench preferences, obtained using the autointerpretation procedure of Movva et al. [2025]. Coefficients are from a linear classifier trained on SAE activations to predict pairwise preferences; prevalence indicates the fraction of pairs in which the feature is active.

Model	Accuracy	Avg. Expl. Len.
claude-3-5-haiku	0.675	292.4
claude-3-7-sonnet	**0.731**	280.2
gpt-4.1	0.702	202.3
gpt-4.1-mini	0.630	246.7
o4-mini	0.700	131.5
deepseek-v3	0.700	167.4
deepseek-r1	0.710	142.8
gemma-3-12b-it	0.657	497.0
llama-3.1-8b	0.581	332.0
qwen-2.5-7b	0.599	174.0

Table 3: LLM-as-a-judge evaluation results by model on LitBench.

highest accuracy.

5. Append the standardized output format instruction, depending on the model size and capacity to follow instructions.

A.4 Prompt Templates.

1. Writer-ly Criteria

Writer-ly Criteria Prompt

```
You're evaluating creative writing
responses A and B.

Compare them based on these dimensions:
- Imagery: vivid descriptions and sensory
details
- Tension: dramatic interest and conflict
- Pattern: structural elements and
composition
- Energy: engaging style and dynamic
writing
- Insight: meaningful ideas and depth

IMPORTANT: Your answer MUST use EXACTLY
this format:
Reasoning: [brief comparison]
Preferred: [A or B] (state which one is
better)

Example format:
Reasoning: Response B has stronger imagery
and tension.
Preferred: B
```

2. Alternative Criteria

Alternative Criteria Prompt

```
Evaluate creative writing responses A and
B.

Consider these aspects:
- Originality: unique concepts,
unexpected elements
- Imagery: sensory language and
descriptions
- Emotional impact: how the writing
affects the reader
- Coherence: logical flow and narrative
structure
- Technical skill: language use and style

FORMAT REQUIRED:
Reasoning: [your evaluation]
Preferred: [A or B]
```

3. Minimal Instruction

Minimal Instruction Prompt

```
Compare responses A and B for creative
writing quality.
MUST follow this format:
Reasoning: [brief analysis]
Preferred: [A or B]
```

4. Reddit-Minimal

Reddit Minimal Instruction Prompt

```
You are evaluating two creative writing
responses (A and B) to the same writing
prompt.
Your task is to predict which response
would receive more upvotes from the Reddit
community.

Your verdict MUST follow this exact
format:
Reasoning: [explain which response would
likely get more Reddit upvotes and why]
Preferred: [A or B] (the one you predict
would get more upvotes)
```

5. Reddit-Verbose

Reddit-Verbose Prompt

```
You are evaluating two creative writing
responses (A and B) to the same writing
prompt. These responses are similar to
those posted on Reddit writing subreddits
like r/WritingPrompts.

Your task is to predict which response
would receive more upvotes from the Reddit
community. Reddit users typically upvote
creative writing that is engaging,
original, well-written, and emotionally
resonant.

When making your prediction, consider what
makes content popular on Reddit:
- Originality and uniqueness of ideas
- Engaging narrative style and pacing
- Emotional impact and relatability
- Clever twists or satisfying conclusions
- Technical quality of writing

This is an experiment to test how well
language models can predict human
preferences in creative writing as
expressed through Reddit's voting system.

Your verdict MUST follow this exact
format:
Reasoning: [explain which response would
likely get more Reddit upvotes and why]
Preferred: [A or B] (the one you predict
would get more upvotes)
```

6. Reddit-Verbose Permuted

Reddit-Verbose Permuted

```
You are tasked with evaluating two
creative writing responses (A and B) to
the same prompt. Your goal is to predict
which response would garner more upvotes
from the Reddit community, specifically in
writing subreddits like r/WritingPrompts.

Consider the following key dimensions for
your evaluation:
- Creativity and Originality: How unique
are the ideas presented?
- Narrative Engagement: Is the
storytelling captivating and immersive?
- Emotional Resonance: Does the piece
evoke feelings or relatable experiences?
- Surprise and Satisfaction: Are there
clever twists or fulfilling conclusions?
- Writing Quality: Is the grammar, style,
and structure polished?

Your output must strictly follow this
format:
1. Reasoning: [Explain which response is
likely to receive more Reddit upvotes,
citing specific strengths and weaknesses.]
2. Preferred: [A or B]

Be concise and clear in your assessment,
adhering to the format above.
```

A.5 Dataset Licenses and Access

All data is publicly accessible via the SAA-Lab/LitBench collection on Hugging Face. Code to use this dataset is available on GitHub.

Our train set content is sourced from euclaise/WritingPrompts_preferences on Hugging Face, which has an MIT-license. In our test set, we release ids of 3.5k Reddit comments from `r/WritingPrompts`, along with code to rehydrate from the reddit api. We acknowledge that Reddit users retain copyright over their individual comments, and we do not claim ownership or offer any re-licensing of this content. We contacted Reddit in advance of this release to clarify acceptable use under their API terms. As of submission time, we have not received a response.

A.6 Chain of Thought Analysis

Figure A.6 gives a full textual analysis of the content within CoT and how predictive they are of prediction accuracy.

A.7 Compute Usage

All training runs and evaluation was done on our internal cluster using a node with 128 CPU cores, 8 NVIDIA A40 GPUs each with 48GB of VRAM, and a total of 732GB of system RAM. Training verifiers took between 3 hours for 1B-parameter models and up to one day for 8B parameter models. The total compute used, including failed runs, data ablations, and generation for LLM-as-a-judge is estimated at 500 GPU-hours on NVIDIA A40.

A.8 Evaluation on Edit-Based Writing Benchmarks

For completeness, we evaluate our best LitBench-trained reward model on the public writing-quality benchmarks released by Chakrabarty et al. [2025a]. These benchmarks evaluate preferences over edited drafts, largely on LLM-generated text, and differ substantially from LitBench, which focuses on story-level preferences over human-written narratives. Our model achieves the following accuracies on their benchmarks: Synthetic-Mirror: 50.18% (563/1122), Style-Mimic: 66.78% (201/301), LAMP-Test: 47.77% (578/1210), Art-or-Artifice: 47.22% (68/144), LM Arena: 53.75% (1053/1959).

A.9 Training Hyperparameters

For all training runs, we use an effective batch size of 128 examples, a learning rate of 1e-5 with a warmup ratio of 10%. We train in `bfloat16` and use AdamW as our optimizer.

A.10 Human Annotation Details

We recruit participants via Prolific and ensure a $12 per hour rate for each rater. The instruction and comprehension stages are show in Figure 9 and 10.

References

Danial Alihosseini, Ehsan Montahaei, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 90–98, Minneapolis, MN. Association for Computational Linguistics.

Teresa M. Amabile. 1982. Social psychology of creativity: A consensual assessment technique. *Journal of Personality and Social Psychology*, 43(5):997–1013.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

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

Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. 2024. Art or artifice? large language models and the false promise of creativity. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–34.

Tuhin Chakrabarty, Philippe Laban, and Chien-Sheng Wu. 2025a. Ai-slop to ai-polish? aligning language models through edit-based writing rewards and test-time computation. *Preprint*, arXiv:2504.07532.

Tuhin Chakrabarty, Philippe Laban, and Chien-Sheng Wu. 2025b. Can ai writing be salvaged? mitigating idiosyncrasies and improving human-ai alignment in the writing process through edits. *Preprint*, arXiv:2409.14509.

John Joon Young Chung, Vishakh Padmakumar, Melissa Roemmele, Yuqian Sun, and Max Kreminski. 2025. Modifying large language model post-training for diverse creative writing. *Preprint*, arXiv:2503.17126.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022a. Understanding dataset difficulty with $\mathcal{V}$-usable information. *arXiv preprint arXiv:2110.08420*.

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

Angela Fan, Mike Lewis, and Yann Dauphin. 2018a. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018b. Hierarchical neural story generation. *Preprint*, arXiv:1805.04833.

Benjamin Feuer, Micah Goldblum, Teresa Datta, Sanjana Nambiar, Raz Besaleli, Samuel Dooley, Max Cembalest, and John P. Dickerson. 2025. Style outweighs substance: Failure modes of llm judges in alignment benchmarking. *Preprint*, arXiv:2409.15268.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, and 1 others. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul

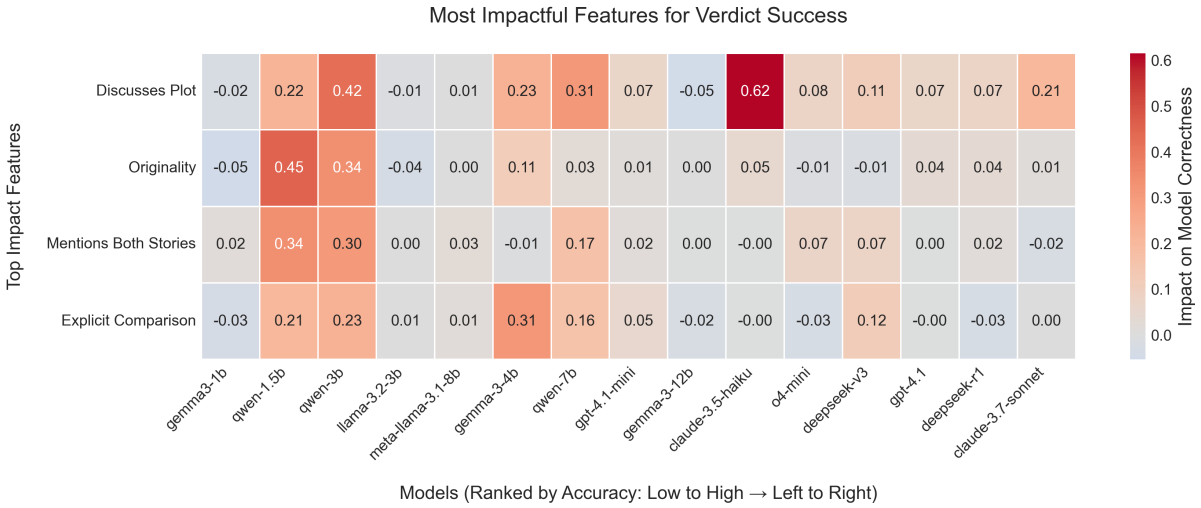

Figure 8: Qualities of explanation text that impact verdict accuracy.

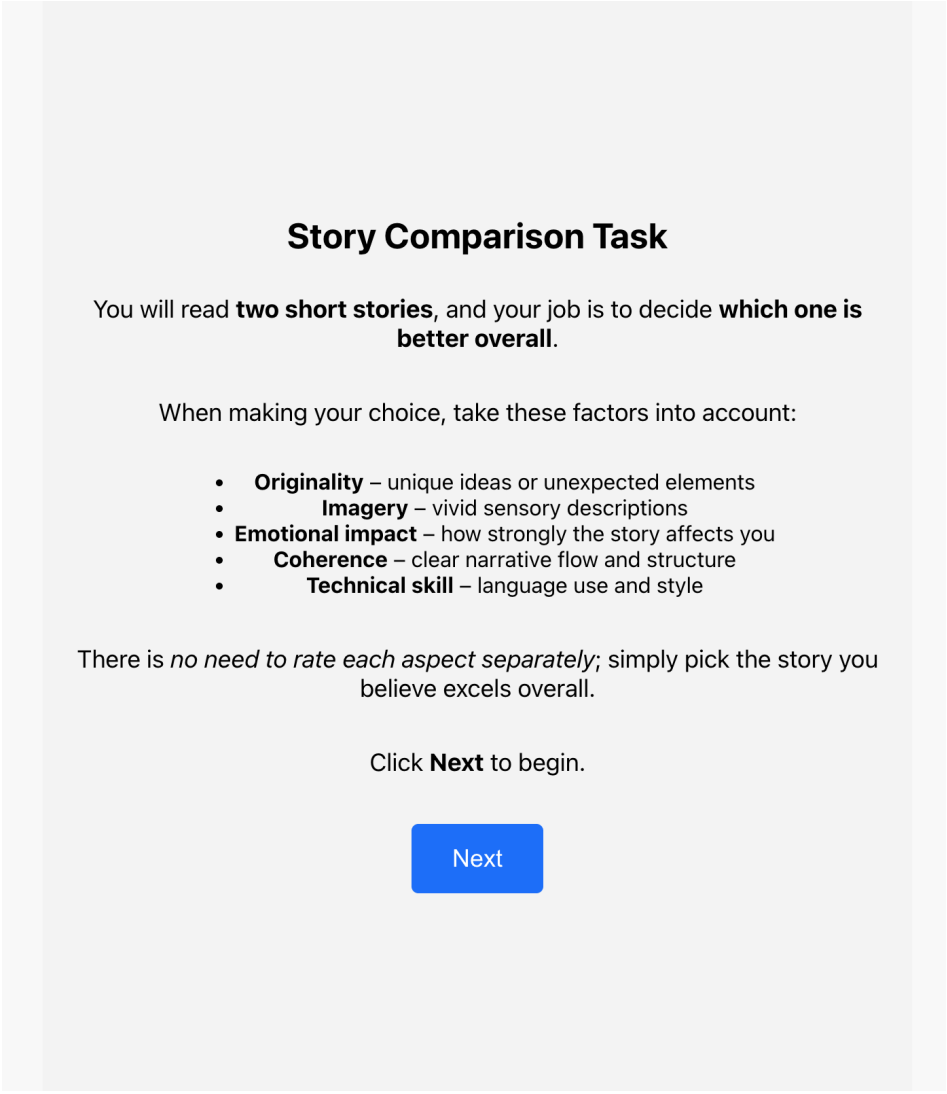

Figure 9: Interface for human annotations - introduction.

Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

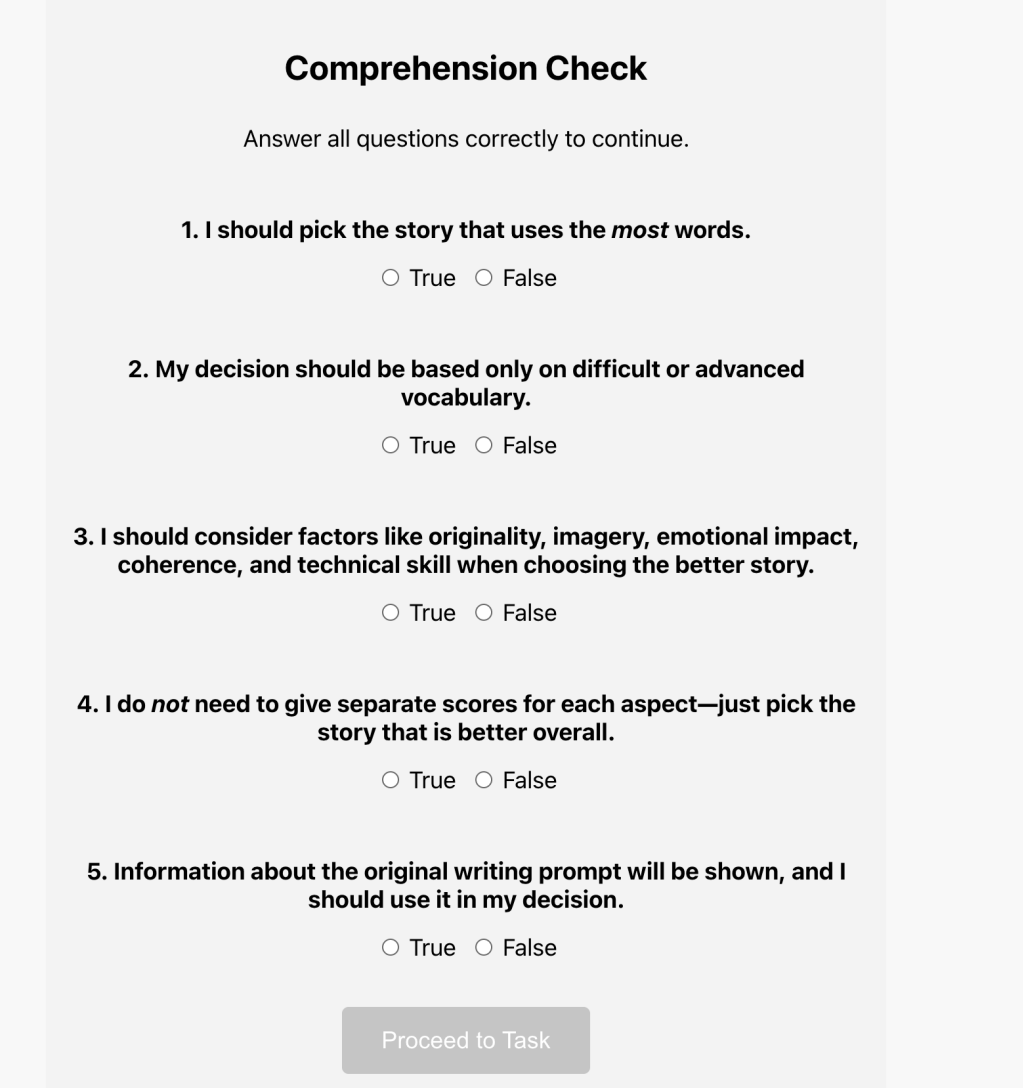

Figure 10: Interface for human annotations - comprehension test.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, and 1 others. 2024. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*.

Jianing Li, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2020. On the relation between quality–diversity evaluation and distribution-fitting goal in text generation. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5927–5937. PMLR.

Ruizhe Li, Chiwei Zhu, Benfeng Xu, Xiaorui Wang, and Zhendong Mao. 2025. Automated creativity evaluation for large language models: A reference-based approach. *Preprint*, arXiv:2504.15784.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Yang Liu, Jonas Schneider, Jonathan Raiman, Ian Tenney, Nitish Gupta, Diya Raghu, Douwe Kiela, and Lazaros Polymenakos. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. 2024. Generative reward models. *arXiv preprint arXiv:2410.12832*.

Rajiv Movva, Smitha Milli, Sewon Min, and Emma Pierson. 2025. What's in my human feedback? learning interpretable descriptions of preference data. *Preprint*, arXiv:2510.26202.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Samuel J. Paech. 2024. Eq-bench: An emotional intelligence benchmark for large language models. *Preprint*, arXiv:2312.06281.

Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. 2024. Training software engineering agents and verifiers with swe-gym. *arXiv preprint arXiv:2412.21139*.

Heather Sellers. 2021. *The Practice of Creative Writing: A Guide for Students*, 4 edition. Bedford/St. Martin's (Macmillan Learning), New York, NY.

Guijin Son, Hyunwoo Ko, Hoyoung Lee, Yewon Kim, and Seunghyeok Hong. 2024. Llm-as-a-judge and reward model: What they can and cannot do. *Preprint*, arXiv:2409.11239.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. *arXiv preprint arXiv:2009.01325*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023a. Large language models are not fair evaluators. *Preprint*, arXiv:2305.17926.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2023b. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*.

Hui Wei, Shenghua He, Tian Xia, Fei Liu, Andy Wong, Jingyang Lin, and Mei Han. 2025. Systematic evaluation of llm-as-a-judge in llm alignment tasks: Explainable metrics and diverse prompt templates. *Preprint*, arXiv:2408.13006.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, and Fei Huang. 2025. Writingbench: A comprehensive benchmark for generative writing. *Preprint*, arXiv:2503.05244.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V Chawla, and Xiangliang Zhang. 2024. Justice or prejudice? quantifying biases in llm-as-a-judge. *arXiv preprint arXiv:2410.02736*.

Mert Yuksekgonul and 1 others. 2025. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639:609–616.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024. Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240*.

Lianmin Zheng and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.