Through the Judge's Eyes: Inferred Thinking Traces Improve Reliability of LLM Raters

Anonymous authors
Paper under double-blind review

Abstract

Large language models (LLMs) are increasingly used as raters for evaluation tasks. However, their reliability is often limited for subjective tasks, when human judgments involve subtle reasoning beyond annotation labels. **Thinking traces**, the reasoning behind a judgment, are highly informative but challenging to collect and curate. We present a human-LLM collaborative framework to infer thinking traces from label-only annotations. The proposed framework uses a simple and effective rejection sampling method to reconstruct these traces at scale. These inferred thinking traces are applied to two complementary tasks: (1) fine-tuning open LLM raters; and (2) synthesizing clearer annotation guidelines for proprietary LLM raters. Across multiple datasets, our methods lead to significantly improved LLM-human agreement. Additionally, the refined annotation guidelines increase agreement among different LLM models. These results suggest that LLMs can serve as practical proxies for otherwise unrevealed human thinking traces, enabling label-only corpora to be extended into thinking–trace–augmented resources that enhance the reliability of LLM raters.[1]

1 Introduction

Large language models (LLMs) are increasingly used as automated evaluators for open-ended text generation tasks, primarily due to their remarkable efficiency and scalability (Gu et al., 2024a; Zheng et al., 2023). This approach has been applied to evaluate diverse applications such as code generation (Zhou et al., 2025), text summarization (Bedemariam et al., 2025), and dialogue (Gu et al., 2024a). However, the reliability of LLM evaluators often decreases in subjective tasks that require nuanced human judgment (Ismayilzada et al., 2024; Gómez-Rodríguez and Williams, 2023), especially when they did not receive specialized training (Krumdick et al., 2025).

To understand what is missing in calibrating these models, we can draw an analogy from the process of training human annotators for qualitative coding tasks. To ensure alignment on a subjective task, a novice annotator typically benefits from three key components that an expert can provide: (1) an *annotation codebook* with detailed instructions and rubrics; (2) a set of *examples* illustrating representative inputs and their corresponding labels; and critically, (3) *explanations* detailing the reasoning for assigning a specific label to each example. Among the three components, explanations are particularly important in creating a shared understanding of the task among annotators (Artstein and Poesio, 2008; O'Connor and Joffe, 2020). We hypothesize that these explanations of the expert's reasoning, which we term the **thinking traces**, are equally necessary to calibrate LLM evaluators and improve their alignment with human judgment.

Despite their value, thinking traces are largely absent from existing annotation datasets. The primary reason is the substantial effort required: articulating and recording a detailed reasoning process is significantly more time-consuming and costly than providing a single label (DeYoung et al., 2019). Most annotation interfaces are not designed to capture this information, and incentivizing raters to produce high-quality traces is challenging (Carton et al., 2020). As a consequence, many datasets for subjective tasks are limited to sparse and sometimes unreliable labels, lacking the rich context that thinking traces would provide.

[1]For reproducing results, our **anonymous codebase** can be accessed at: `https://anonymous.4open.science/r/thru_judge_eye-56DD/`

Please read the prompt, the human story and the subject story (both stories might be the same). The story you will have to rate is the subject story.

Note: some stories have been abruptly cut in the middle of a sentence. Please rate them as if they ended just before the unfinished sentence.

Note: if the story is not relevant with respect to the prompt, it only affects the Relevance criterion! Do not rate 1 everywhere.

Then, please rate the subject story on a scale from 1 (worst) to 5 (best) on Complexity:

1: The story is very simple with no elaborate elements.
2: The story is somewhat simple with few elaborate elements.
3: The story has some complexity but lacks depth in certain areas.
4: The story is mostly complex with minor areas lacking depth.
5: The story is highly complex and elaborate throughout.

Figure 1: An example of the annotation codebook for evaluating complexity of short stories (Chhun et al., 2022). Only basic instructions and vague scoring rubrics are provided.

Reasoning language models (RLMs) offer a promising opportunity to address this issue (Comanici et al., 2025; Guo et al., 2025; Jaech et al., 2024). These models are designed to generate step-by-step reasoning, often termed "intermediate reasoning tokens", before producing a final answer (Guo et al., 2025). This model-generated reasoning is intended to align with the underlying thinking traces of human experts, also known as "human priors" in Guo et al. (2025). If this alignment is faithful, these models could serve as a proxy to reconstruct thinking traces at scale, mitigating the costly human annotation process. However, whether AI-generated reasoning truly aligns with human reasoning remains an open question, especially for subjective tasks where reasoning can be highly nuanced. While directly verifying this alignment is challenging, given the lack of ground-truth human thinking traces, a more pragmatic research question in our context is: **Can these model-inferred thinking traces, even if imperfect, serve as a useful proxy to improve the reliability of automated LLM raters?**

To tackle this question, we present a human-LLM collaborative framework to infer thinking traces from label-only human annotations. We employ a simple, yet effective, rejection sampling method to reconstruct these traces at scale. We demonstrate the utility of inferred thinking traces through two complementary applications. First, we use them as training data to fine-tune open-weight LLM raters so that they are specialized for the evaluation task. Second, for models where fine-tuning is not an option, we introduce a novel method for codebook refinement. Inspired by iterative refinement practices in qualitative coding (O'Connor and Joffe, 2020), this approach addresses the common problem of ambiguous instructions in many annotation codebooks (Klie et al., 2023) (Figure 1). Our automatic pipeline leverages the inferred traces to synthesize a new, more explicit codebook that better steers the model's judgments. Experiments show that both methods significantly improve the alignment of LLM raters with human judgments. Furthermore, the synthesized codebook also improves the agreement between different LLM providers, demonstrating its utility in promoting cross-model consistency. These findings demonstrate that model-inferred thinking traces can effectively serve as a practical proxy for human reasoning, thus improving the reliability and consistency of automated LLM raters on subjective evaluation tasks.

2 Related Work

Reasoning Language Models. Reasoning Language Models (RLMs) represent a significant evolution from standard LLMs, as they are specifically trained to solve tasks that require multiple steps of deliberation. This class of models has become the de facto standard for high performance (Comanici et al., 2025; Guo et al., 2025; Jaech et al., 2024), demonstrating superior capabilities in tasks such as math and coding. Models such as DeepSeek R1 demonstrate the effectiveness of large-scale reinforcement learning in cultivating complex reasoning behaviors in these models (Guo et al., 2025), and provide public access to these reasoning tokens. Although these models are intended to align with "human priors" (Guo et al., 2025), the validity of these

Okay, let's tackle this rating. The user wants me to evaluate the Subject Story based on the Engagement criterion. The prompt mentions that stories create actual universes, and unfinished ones are hellish. The Human Story talks about a life controlled by a creator, leading to existential dread when stories are incomplete.

Now, looking at the Subject Story. It starts with a character exhausted from playing games, going home to feed his family. Then there's a sudden shift to a genie game over the ocean, hot water leaking, cold sea shattering the window. The narrative seems disjointed. The protagonist has a baseball bat, gets hit in the head, a storm hits, and the story ends abruptly.

Engagement-wise, the story is confusing. The transitions between scenes are jarring. The mix of mundane elements (family, baseball bat) with surreal events (genie, sudden storms) might intrigue some readers but could also lose them. The abrupt ending, while fitting the prompt's theme of unfinished stories, might leave readers frustrated. The imagery is vivid but lacks coherence, making it hard to follow. The emotional build-up from hope to despair isn't clearly developed here. The reader might struggle to connect with the protagonist or the plot. So, Engagement is probably low. I'd rate it a 2 because it has some interesting elements but is difficult to follow and loses interest quickly.

Figure 2: An example of the inferred thinking trace in evaluating the engagement level of a short story.

reasoning tokens remains a debatable question (Amirizaniani et al., 2024; Dasgupta et al., 2024; Mondorf and Plank, 2024). Testing the validity of thinking traces is beyond the scope of this paper. However, we are interested in a practical objective: leveraging RLMs as a tool to infer latent thinking traces from human experts, thereby improving the reliability of downstream LLM raters.

Distilling Reasoning via Rejection Sampling. A powerful paradigm for generating synthetic data is *rejection sampling fine-tuning*, where a model generates numerous candidate reasoning traces, and only those passing a filtering criterion are used for subsequent training. Existing work has pioneered this approach for tasks with objectively correct answers, such as mathematical reasoning, using an automatic verifier to accept only correct outputs (Guo et al., 2025; Zelikman et al., 2022; Wadhwa et al., 2024; Tong et al., 2024; Singh et al., 2023). To handle more nuanced tasks, methods such as RAFT (Dong et al., 2023) replaced the binary verifier with a learned reward model, filtering for outputs that score highly on desired attributes. This trend of applying reasoning to more subjective domains is also evident in recent work on recommender systems (Tsai et al., 2024). Crucially, while previous work exclusively focuses on using these curated traces for fine-tuning, we introduce a novel second application: distilling insights from the thinking traces to refine annotation codebooks automatically.

Refining Codebooks for Qualitative Coding and Annotation The refinement of codebooks is a long-standing challenge in both qualitative research and large-scale data annotation. Research in HCI and crowdsourcing communities developed human-centric workflows to improve task instructions, often relying on crowd workers to identify ambiguities that were then resolved by an expert or through discussion (Chang et al., 2017; Manam and Quinn, 2018; K. Chaithanya Manam et al., 2019; Pradhan et al., 2022). More recently, researchers have begun to leverage LLMs to automate this process. This includes work on human-LLM collaboration for *qualitative coding* (Xiao et al., 2023; Halterman and Keith, 2025; Wiebe et al., 2025; Torii et al., 2024; Meng et al., 2024). Other work has demonstrated that LLMs can automatically analyze data or synthesize new guidelines to improve downstream performance (Bibal et al., 2025; Hsu et al., 2025; Srivastava et al., 2025). A key distinction of these automated methods is that they rely only on an LLM's intrinsic capabilities to analyze content or generate text, rather than being grounded in direct human annotation feedback. In contrast, our approach creates a collaborative pipeline with inputs from both humans and LLMs.

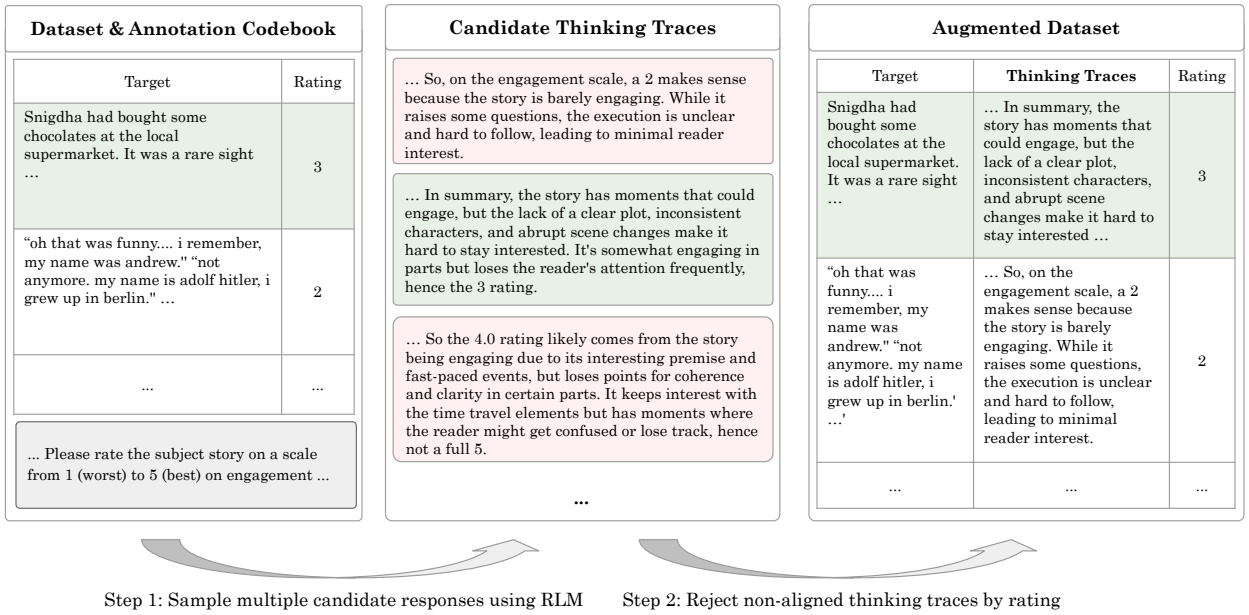

Figure 3: Illustration of Inferring Thinking Traces through an RLM. Multiple candidate thinking traces and labels are sampled. Only candidates that are aligned with human-annotated labels are preserved. The detailed process is provided in Section 3.1.

3 Method

3.1 Inferring Latent Thinking Traces

Our method begins with a seed dataset of human-annotated examples, denoted as $\mathcal{D}_{\text{human}} = \{(x_i, y_i)\}_{i=1}^{N}$. Here, x_i represents an annotation target (e.g., a story) and y_i is the corresponding label (e.g., a Likert rating for story 'complexity') assigned by human raters based on a specific codebook $\mathcal{C}$.

For any non-trivial annotation task, a human rater engages in a cognitive process to arrive at the final label y_i. We formalize this unobserved process as the human's **latent thinking trace**, t_i. This trace, which includes applying guidelines, resolving ambiguities, and weighing evidence, is not recorded in standard annotation workflows due to the high cost and inherent difficulty of articulating such complex reasoning.

Our goal is to generate a high-quality proxy for this latent trace, which we term the **reconstructed thinking trace**, t'_i. To achieve this, we use an RLM that can generate thinking tokens, hereafter referred to as the *generator model*, to perform rejection sampling. For each input item x_i, we provide the generator model with the codebook $\mathcal{C}$ and the annotation target and prompt it to generate a step-by-step thinking trace that concludes with a final rating. This generation is performed k times independently, yielding a set of k candidate trace-label pairs, $\{(t'_{i,j}, y'_{i,j})\}_{j=1}^{k}$ where $t'_{i,j}$ denotes the j-th candidate trace and $y'_{i,j}$ denotes the j-th predicted label for x_i.

For a model to reach the correct human label y_i, its reasoning is more likely (but not guaranteed) to be a plausible approximation of the latent human thought process. Therefore, we filter the candidate set by retaining only those traces where the model's predicted label $y'_{i,j}$ matches the ground-truth human label y_i. This creates a set of traces for each sample x_i:

$$T'_i = \{t'_{i,j}, j \in \{1, \ldots, k\} \mid y'_{i,j} = y_i\}$$

From this set T'_i, we select the first sample to serve as the final reconstructed thinking trace, t'_i. This transforms our initial dataset into an augmented, reasoning-rich dataset, $\mathcal{D}_{\text{reason}} = \{(x_i, t'_i, y_i)\}_{i=1}^{N}$, which forms the basis for the alignment techniques described next. In essence, each data point in $\mathcal{D}_{\text{reason}}$ can be considered a product of human-LLM collaboration, where the human label is used as a verifier to select an

LLM-generated thinking trace. An example inferred thinking trace is shown in Figure 2 and an illustration of the overall process is provided in Figure 3.

3.2 Improving LLM Raters

The primary goal of inferred thinking traces, $\mathcal{D}_{\text{reason}}$, is to improve the reliability of LLM raters. We show our method's utility in two complementary scenarios, which cover the primary ways LLMs are used in practice: as fine-tunable models that can be specialized to specific domains; and as black-box APIs. For open-weight models, we use $\mathcal{D}_{\text{reason}}$ to directly fine-tune the LLM rater, aligning its judgments with human reasoning patterns. More broadly, for the common case of proprietary models, we instead use our dataset to automatically refine the annotation codebook. This refined codebook serves as a more effective prompt to guide the model's behavior. This offers a practical, training-free way to leverage state-of-the-art LLMs.

Fine-Tuning Specialized LLM Raters. We show that the augmented dataset $\mathcal{D}_{\text{reason}}$ provides a powerful training resource for creating a specialized LLM rater. Unlike standard supervised fine-tuning (SFT), which only trains on the input-label pairs (x_i, y_i), the augmented dataset provides the model with the complete reasoning trace t'_i. We hypothesize that the thinking traces contribute to a much richer training signal, which can significantly improve the model's alignment with human feedback. The model is trained using a standard SFT objective, as shown in Equation 1, to predict the full thinking trace followed by the final label. Here, π_θ denotes the LLM rater with weights θ. This objective teaches the model not only *what* label to produce, but also *how* to reason towards it.

$$\mathcal{L}_{\text{SFT}} = \mathbb{E}_{(x,t',y)\sim\mathcal{D}_{\text{reason}}} \left[-\log \pi_\theta(t', y|x) \right] \tag{1}$$

Refining Annotation Codebooks. We further investigate ways to take advantage of the inferred thinking traces when using proprietary LLMs whose weights are unavailable (i.e., they cannot be updated). Our approach is motivated by the observation that many codebooks contain ambiguous descriptions or lack concrete criteria, resulting in inconsistent ratings when used directly as prompts, as shown in Figure 1. Our collection of reconstructed traces serves as a corpus of successful step-by-step applications of these guidelines. We can leverage these traces to improve the codebook by extracting common reasoning patterns and synthesizing a more explicit step-by-step procedure. This improved codebook, $\mathcal{C}'$, can then be used to construct more effective prompts for proprietary LLMs where fine-tuning is not an option.[2]

Specifically, we propose a two-stage process that targets the two main components of the codebook: the *task instructions* and the *scoring rubric*.

1. *Improve task instructions*: We first sample 10 thinking traces for each rating level and prompt an LLM to summarize the common reasoning patterns in these traces into an explicit, step-by-step procedure for the new task instructions.

2. *Enrich scoring rubrics:* To enrich the scoring rubric, we then sample a set of 50 thinking traces for each rating level and extract short critiques from each one (e.g., "The lack of a coherent plot…"). Not all critiques are representative of a given rating level. For example, a good story may still have some flaws. To find the most representative critiques, we cluster the text embeddings of these critiques using the `text-embedding-3-large` model and select the critiques with the highest semantic similarity to each cluster's centroid. Finally, we present these representative critiques to an LLM, prompting it to synthesize a new, detailed rubric for each rating level that is grounded in concrete examples.

We provide the complete prompts for this process in Appendix C.

[2]While we do not test this hypothesis in our work, an improved codebook could also potentially enhance the consistency and quality of human raters.

Read the Materials Thoroughly Start by reading the prompt, the human story, and the subject story. Note that the subject story is the one being rated, not the human story ...

Step-by-Step Rating

- Look for key elements such as characters, events, plot, themes, or setting ...
- Evaluate if the identified elements are developed, interconnected, and coherent ...
- Consider whether the story follows a structured progression (linear or non-linear) ...
- Stories with minimal elements or undeveloped content are simpler. Stories that attempt multi-layered exploration of ideas, even if not fully realized, should be evaluated as more complex.

Select the Rating

1. The story is very simple, lacks depth, and introduces no intricate or developed elements.
2. The story is somewhat simple, with few elements that are underdeveloped, fragmented, or disconnected.
3. The story demonstrates some complexity, introducing multiple elements, but lacks depth, development, or coherence in integrating these elements.
4. The story is mostly complex, weaving together several developed elements with minor areas that could be expanded or refined.
5. The story is highly elaborate and complex, integrating multiple layers (e.g., detailed world-building, character depth, advanced structure, thematic sophistication) in a cohesive and interconnected way.

Write the final rating Place your chosen rating ...

Figure 4: An example of the refined annotation codebook. Some content has been omitted due to space limitations. See Appendix D for a complete example.

3.3 Evaluation

To empirically validate LLM raters, we use two primary criteria. The first, *Agreement with Human Judgments*, serves as the main test for both applications. The second, *Inter-Rater Reliability*, is an assessment designed to measure the specific impact of our codebook refinement. All evaluations are performed on held-out test sets randomly sampled from the original dataset.

Agreement with Human Judgments This criterion measures the degree to which an LLM rater's outputs agree with human judgments. We apply this evaluation to the outputs of both our applications. For this analysis, we treat a held-out set of human ratings as the gold standard and compare the LLM-generated labels against them. To quantify this agreement, we use several standard metrics for labels in the Likert scale. Our primary focus is on **Kendall's Tau (τ)**, as ranking correlation[3] is the most used measure in the literature (Liu et al., 2023; 2024; Thakur et al., 2025; Gu et al., 2024b). For a more complete assessment, we also report the Mean Squared Error (MSE) for average error magnitude and the Intraclass Correlation Coefficient (ICC3) for consistency, based on a two-way mixed-effects model (Koo and Li, 2016).

Inter-Rater Reliability. For codebook refinement, another robust indicator of success is whether $\mathcal{C}'$ enables independent raters to arrive at the same conclusions consistently. High inter-rater reliability is a sign of a clear and well-defined annotation process. To measure this, we use the improved codebook $\mathcal{C}'$ to prompt M different proprietary LLMs to rate the same set of items. We then calculate the **reliability across these M independent LLM raters** using ICC3 (Koo and Li, 2016).

Statistical Significance Test. To assess statistical significance, we use the following statistical tests. For fine-tuned LLM raters, we apply a paired t-test on MSE, and employ one-sided bootstrap hypothesis tests to evaluate whether improvements in τ and ICC3 are statistically significant in the direction of superiority. For

[3]We observe a high correlation between Spearman's ρ and Kendall's τ in the results. We only report τ here for simplicity.

the refined codebook analysis, we conduct one-sided paired t-tests on each metric, comparing the performance of four LLM raters before and after refinement, thereby directly testing whether the refinement leads to consistent improvements across raters. To assess the statistical significance of agreement between LLM raters, we conduct a one-sided bootstrap hypothesis test to evaluate whether the improvement in ICC3 is significant in the direction of superiority.

4 Experiment

Task	**Original Model**			**SFT (DeepSeek Trace)**			**SFT (gpt-oss Trace)**		
Metric	**ICC_3**	**τ**	**MSE**	**ICC_3**	**τ**	**MSE**	**ICC_3**	**τ**	**MSE**
Short Story (Complexity)	0.231	0.177	2.016	0.604*	0.463*	0.838*	0.581*	0.460*	0.944*
Short Story (Engagement)	0.289	0.275	1.939	0.364	0.048	1.364	0.371	0.179	1.831
Student-Written Essay	0.233	0.202	1.947	0.267	0.322*	1.149*	0.244	0.240	1.387*
News Summary (Consistency)	0.186	0.141	2.904	0.319	0.276	3.476	0.331	0.243	2.970
News Summary (Fluency)	0.378	0.275	1.938	0.402	0.319	1.160*	0.498	0.306	1.200*
Translation	0.240	0.171	2.778	0.348*	0.260*	2.492	0.314*	0.256*	2.439*
Chatbot Response (Correctness)	0.288	0.162	2.049	0.403*	0.297*	1.815*	0.433*	0.263*	1.897
Chatbot Response (Helpfulness)	0.274	0.176	1.969	0.422*	0.264*	1.723*	0.466*	0.295*	1.575*
Average	**0.265**	**0.197**	**2.193**	**0.391**	**0.281**	**1.752**	**0.405**	**0.280**	**1.780**

Table 1: Effect of Reasoning-Enhanced SFT on Human-LLM Rater Agreement. For the metrics, higher is better for ICC_3 and Kendall's τ, while lower is better for MSE. An asterisk (*) denotes a statistically significant improvement over the baseline ($p < 0.05$).

4.1 Datasets

We evaluated our framework across five diverse annotation tasks, all of which require deliberate reasoning rather than simple intuition to assign a reliable score. Another key selection criterion is the public availability of the annotation codebook used for each task. These tasks cover two distinct types of human feedback on generated text: single holistic quality scores and fine-grained scores across multiple dimensions. For consistency in our evaluation methodology, we focused on datasets that use a Likert scale. However, our framework is readily applicable to other formats, such as forced-choice rankings. The five tasks are summarized below:

1. **Evaluating Short Stories.** We use the HANNA dataset (Chhun et al., 2022), an annotation dataset for evaluating machine-generated stories. The stories were rated by human annotators recruited through Amazon Mechanical Turk, restricted to native English speakers with a Master's Qualification. The *complexity* dimension provides a measure of structural and elemental richness, while *engagement* captures readers' emotional involvement, reflecting a more subjective perception.

2. **Evaluating Student-Written Essays.** For this task, we utilize a dataset of student essays from a Kaggle competition on automated essay scoring (Crossley et al., 2024). Each essay is assigned a single *holistic* score by expert human graders.

3. **Evaluating News Summaries.** We use the SummEval benchmark (Fabbri et al., 2020), which provides human evaluations of machine-generated summaries of news articles. Each summary is annotated by three expert annotators with prior experience in writing research paper summaries for academic conferences. We focus on the *consistency* and *fluency*, which respectively evaluate factual correctness and linguistic quality, two key aspects of summary reliability and readability.

4. **Evaluating Translations.** The WMT 2020 dataset (Freitag et al., 2021) provides expert human evaluations of machine translation outputs under the Scalar Quality Metric framework. We focus on the Chinese-to-English translation task, where each translation is given a *holistic* quality score by three professional translators native in the target language.

5. **Evaluating Chatbot Responses.** We use the HelpSteer2 dataset (Wang et al., 2024), which contains human feedback on AI-generated conversational responses. The annotations were collected from trained human raters on the Scale AI platform, where each response was evaluated by at least three annotators across five quality dimensions. Additional annotators were assigned when initial ratings showed high disagreement to ensure reliability. We focus on *helpfulness*, which measures informativeness and relevance of the response, and *correctness*, which evaluates the factual accuracy.

More statistics and pre-processing details for each dataset are provided in Appendix A.

4.2 Data Processing

Filter Gold Standards The evaluation of an LLM rater requires comparing its judgments with a reliable gold standard derived from human consensus. For datasets that already provide a single aggregated human score per sample, such as Essay and HelpSteer2, we adopt this value directly as our gold standard. For datasets that provide raw ratings from multiple annotators, such as HANNA, SummEval, and WMT-2020, we construct the gold standard by first filtering for quality. We discard contentious examples where the standard deviation of human scores exceeds 1.0, and then define the gold standard as the median of the remaining ratings.

Infer Thinking Traces As outlined in Section 3, we use rejection sampling to infer thinking traces from the annotation datasets. The original annotation codebook from each source serves directly as the prompt. For our generator models, we selected `DeepSeek-R1` and `gpt-oss-120b` as they output complete thinking traces. We prompt each model to generate sixteen ($k = 16$) thinking traces for every sample and keep the first trace that aligns with the human annotator's final rating. A sampling temperature of 1.0 is used during inference to encourage generative diversity.

4.3 Fine-Tuning Specialized LLM Rater

This experiment demonstrates how inferred thinking traces can be leveraged to train more effective automated evaluators via reasoning-enhanced supervised fine-tuning.

We use `DeepSeek-R1-0528-Qwen3-8B` as our base model (Guo et al., 2025). The model is then fine-tuned on each individual task separately. During fine-tuning, the model is trained not only to predict the final rating but also to generate the entire thought process. This approach encourages the model to learn the underlying reasoning process that leads to a specific judgment. To prevent overfitting, we use early stopping with a patience of 100 steps on a held-out validation set, selecting the model checkpoint that achieves the highest Kendall's τ correlation with human ratings. The performance of this final model is then evaluated by comparing its ratings against the human gold standard on the test set.

Results and Analysis The results in Table 1 show that reasoning-enhanced SFT significantly improves the performance of the LLM rater. Across all tasks, the models fine-tuned on inferred traces from both `DeepSeek-R1` and `gpt-oss-120b` achieve significant gains over the original model. On average, reasoning-enhanced SFT improves Kendall's τ from 0.197 to 0.281, a relative increase of 42.6%. Notably, the final performance of models trained on traces from `DeepSeek-R1` and `gpt-oss-120b` is highly comparable. Both sets of inferred traces, despite originating from different models, serve as effective training data to improve the base rater. This demonstrates the universality of our framework over heterogeneous RLMs.

An outlier is the *Short Story (Engagement)* task, where both SFT models show a degradation in performance, particularly in Kendall's τ. We hypothesize that this anomaly may stem from a potential lack of sufficient high-quality data points in the seed dataset for this specific dimension, making it difficult to learn a consistent reasoning pattern.

Overall, despite some task-specific variations, the evidence supports that **fine-tuning on inferred thinking traces is an effective method for enhancing the alignment of LLM raters with human judgments**.

Task	Original Codebook			Refined (DeepSeek Trace)			Refined (gpt-oss Trace)		
Metric	**ICC_3**	**τ**	**MSE**	**ICC_3**	**τ**	**MSE**	**ICC_3**	**τ**	**MSE**
Short Story (Complexity)	0.497	0.428	1.116	0.571	0.512	0.634*	0.504	0.401	0.999
Short Story (Engagement)	0.514	0.303	2.044	0.559	0.380	1.525	0.564	0.391*	1.392
Student-Written Essay	0.413	0.421	0.945	0.481	0.491*	0.895	0.433	0.459*	0.957
News Summary (Consistency)	0.150	0.166	2.861	0.178	0.188	2.591	0.181	0.177	2.668
News Summary (Fluency)	0.182	0.142	3.121	0.197	0.192	2.203	0.248	0.190	2.855
Translation	0.419	0.347	2.392	0.448*	0.368	1.942*	0.452	0.356	1.953*
Chatbot Response (Correctness)	0.451	0.354	1.920	0.485	0.370*	1.724*	0.484	0.378*	1.804
Chatbot Response (Helpfulness)	0.475	0.366	1.801	0.511*	0.390*	1.642*	0.484	0.376	1.585*
Average	**0.388**	**0.316**	**2.025**	**0.429**	**0.361**	**1.645**	**0.419**	**0.341**	**1.777**

Table 2: Effect of Codebook Refinement on Human-LLM Rater Agreement. Each value in the table is an average of the corresponding metrics calculated for each LLM rater.

Task	Original Codebook	Refined (DeepSeek Trace)	Refined (gpt-oss Trace)
Short Story (Complexity)	0.613	0.700	0.771*
Short Story (Engagement)	0.883	0.810	0.823
Student-Written Essay	0.560	0.602*	0.554
News Summary (Consistency)	0.405	0.501*	0.417
News Summary (Fluency)	0.352	0.395	0.370
Translation	0.633	0.672*	0.689*
Chatbot Response (Correctness)	0.578	0.649*	0.628*
Chatbot Response (Helpfulness)	0.615	0.636*	0.604
Average	**0.580**	**0.621**	**0.607**

Table 3: Effect of Codebook Refinement on Rater Agreement among LLM Raters (ICC_3).

4.4 Refining Annotation Codebook

Although SFT is effective, it is limited to LLMs for which weights are available to update. In more common cases, we have access to the state-of-the-art model through black-box APIs only. To steer their behavior, we need to synthesize more effective prompts in the same way that we refine the annotation codebook for human raters. Following the procedure described in Sec 3, we generate two refined annotation codebooks for each task using traces from `DeepSeek-R1` and `gpt-oss-120b`, respectively. We test the effectiveness of these refined codebooks on four powerful LLMs from different model providers: `Claude-4-Sonnet`, `Gemini-2.5-Flash`, `GPT-5`, and `DeepSeek V3.1`. We adopt a temperature of 1.0 and sampling probability mass cutoff of 0.95 for inference.

Improved Agreement with Human Judgments. Table 2 shows the **improved average agreement between four LLM raters and human judgments**. On average, the codebooks refined by `DeepSeek-R1` traces improve human-LLM Kendall's τ from 0.316 to 0.361 (a 14.2% improvement), and the codebooks refined by `gpt-oss-120b` traces improve τ to 0.341 (a 7.9% improvement), showing a consistent improvement. Additionally, the ratings obtained by prompting more advanced LLMs with the refined codebook outperform those by fine-tuned smaller LLM raters (in Table 1). This highlights the effectiveness and practicality of refining the annotation codebook.

Improved Inter-Rater Reliability. Crucially, **the refined codebooks also improve inter-rater reliability among different LLM raters**. Table 3 shows that the average inter-rater agreement (ICC_3) increases from 0.580 to 0.621 with the thinking traces inferred by `DeepSeek-R1`. The benefits are most pronounced on tasks with relative objective criteria, such as correctness of *Chatbot Response* and *Translation*. Conversely, on highly subjective tasks like evaluating the engagement of *Short Story*, the refined codebooks can sometimes increase disagreement between LLM raters. However, this is not necessarily a negative indicator: the original codebook may have induced high but superficial agreement by promoting a shared systematic bias among the LLM raters. By providing more nuanced guidance, the refined codebook reduces

this shared bias, causing the models to better reflect the natural and sometimes divergent distribution of human judgments.

Qualitative Analysis of Annotation before and after Refinement. The example in Figure 5 provides a clear illustration of how the refined codebook improves rater alignment. Initially, using the original codebook to evaluate a story (ground truth rating: 2), the two LLM raters produce divergent and incorrect scores. This disagreement stems from two core flaws in the original guide.

First, it lacks a **step-by-step procedure**, leading to inconsistent evaluation paths. Claude, for example, immediately makes a top-down judgment about the story's "lack of structure, coherence, or meaningful plot development." In contrast, DeepSeek takes a bottom-up approach, first identifying "surreal and disjointed aspects" which it views as a source of "complexity." The refined codebook solves this by introducing a clear, standardized process. As seen in the new traces, both raters now begin by systematically identifying the story's elements—such as "narrator routine, home setting, dialogue" (Claude) and "narrator, teacher, child; mundane events" (DeepSeek)—before making a final judgment.

Second, the original **scoring rubric is too abstract**, forcing the models to weigh competing factors differently. The refined rubric corrects this by grounding its descriptions in **concrete examples**. This allows both raters to converge on the same nuanced conclusion. Claude's final justification now states that the story "falls into the category of 'few elements, underdeveloped, fragmented, disconnected,"' while DeepSeek similarly concludes it is "somewhat simple with disconnected elements." They both correctly assign a rating of 2, demonstrating that the refined codebook aligns LLM raters by standardizing the entire evaluation process.

Claude (Before): ... The subject story is unrelated to the prompt. It mostly describes a mundane routine with some disjointed details... Then shifts abruptly to snow and vague dialogue about a "gift" and "danger," but these are underdeveloped. ... The narrative lacks structure, coherence, or meaningful plot development. ... Elements that could add depth are too vague. ... The story reads like disconnected fragments with no complexity. <rating>1</rating>
Claude (After): ... The story includes some elements: narrator routine, home setting, dialogue, and atmosphere. ... But development is weak: abrupt jumps, unexplained characters, disconnected dialogue, no plot or character growth, abrupt ending. ... Though it introduces multiple ideas (routine, relationships, mysterious "gift"), they are fragmented and underdeveloped. ... Falls into the category of "few elements, underdeveloped, fragmented, disconnected." <rating>2</rating>
DeepSeek (Before): ... The subject story is different from the prompt, lacks magical elements. ... Contains surreal and disjointed aspects (time reversing, "gift," ambiguous dialogue) that add some complexity. ... But overall lacks cohesion and depth, feels fragmented. <rating>3</rating>
DeepSeek (After): ... Identifies elements: narrator, teacher, child; mundane events; home/snow setting; vague themes ("gift," "danger"). ... Notes minimal depth, low coherence, abrupt time shifts, unresolved dialogue. ... Structure linear but disjointed. ... Introduces some elements but underdeveloped, no advanced techniques. ... Story is somewhat simple with disconnected elements. <rating>2</rating>

Figure 5: Example thinking traces from two LLM raters on evaluating the complexity of a short story, before and after codebook refinement.

4.5 Ablation Study

Ablation on Refinement Components To understand the individual contributions of our two-stage refinement process, we conduct an ablation study using thinking traces from `DeepSeek R1` on three diverse tasks: *Short Story (Engagement)*, *Chatbot Response (Correctness)*, and *News Summary (Consistency)*. To isolate the effect of each component, we create two separate versions of each codebook: one with only the task instructions refined and another with only the scoring rubric refined, and then assess Human-LLM rater alignment.

Results in Table 4 show that **both instruction refinement and rubric refinement enhance performance, though their relative impact depends on the specific weaknesses of the original codebook**. For the *Chatbot Response* task, instruction refinement yields larger gains, increasing Kendall's τ from 0.354 to 0.374, compared to 0.359 from rubric refinement. In contrast, rubric refinement proves more effective for *News Summary*, raising τ from 0.166 to 0.185, whereas instruction refinement provides only a marginal increase to 0.168. For the *Short Story* task, both refinements deliver comparably strong improvements. This analysis reveals that the optimal refinement strategy depends on the original codebook's deficiencies: rubric refinement is most beneficial when the original lacks a detailed rating scale (as with News Summary), while instruction refinement is critical when it lacks a step-by-step guideline, even with a detailed rubric (as with Chatbot Response). These findings validate our two-stage approach, highlighting its robustness in addressing distinct types of flaws in annotation guidelines.

Task	**Original**			**Refined (Instruction)**			**Refined (Rubric)**			**Refined (Both)**		
Metric	**ICC_3**	**τ**	**MSE**	**ICC_3**	**τ**	**MSE**	**ICC_3**	**τ**	**MSE**	**ICC_3**	**τ**	**MSE**
Story	0.514	0.303	2.044	0.551	0.372	1.867	0.537	0.380	1.701	0.559	0.380	1.525
Chatbot	0.451	0.354	1.920	0.480	0.374	1.827	0.453	0.359	1.836	0.485	0.370	1.724
Summary	0.150	0.166	2.861	0.145	0.168	2.405	0.195	0.185	2.602	0.178	0.188	2.591

Table 4: Ablation study on the effect of refinement codebook component. Story: Short Story (Engagement). Chatbot: Chatbot Response (Correctness). Summary: News Summary (Consistency).

Task	**Original Codebook**			**Refined (Thinking Tokens)**			**Refined (Post Hoc)**		
Metric	**ICC_3**	**τ**	**MSE**	**ICC_3**	**τ**	**MSE**	**ICC_3**	**τ**	**MSE**
Story	0.514	0.303	2.044	0.559	0.380	1.525	0.533	0.330	1.743
Chatbot	0.451	0.354	1.920	0.485	0.370	1.724	0.484	0.376	2.275
Summary	0.150	0.166	2.861	0.178	0.188	2.591	0.160	0.172	2.533

Table 5: Ablation Study on using post hoc explanation.

Ablation on Using Post Hoc Explanations. A key prerequisite to reconstructing effective thinking traces is full access to the RLM thinking tokens. However, this may not always be possible, especially for state-of-the-art model like `gpt-5` and `Gemini-2.5-Pro`, which only expose a brief summary of their internal thinking. Therefore, we investigate whether our framework can be adapted for models that do not expose their internal thinking process, such as most proprietary LLMs. To address this, we test the usage of *post hoc explanations* as an alternative for pre-decision thinking traces. Data collection involves providing a generator model with the annotation target x_i and the gold standard human rating y_i, and prompting it to generate multiple post hoc candidate explanations for how one could arrive at that rating. These candidates are then filtered for quality: we remove all explicit rating information from the generated text and use another LLM to verify if the remaining reasoning still leads to the correct rating y_i. Due to multiple rounds of LLM calls, this method is more costly. The results in Table 5 show that this collection pipeline remains a highly effective approach. The codebook refined using post hoc explanations significantly outperforms the original codebook on both tasks. While its performance is slightly lower than using native thinking traces for the Story task, it is highly comparable for the News Summary (consistency) task. This finding demonstrates the robustness of our framework, suggesting that it can be successfully extended to LLMs without explicit thinking tokens.

5 Discussion

We list several limitations and potential directions that could be addressed in future research:

Sampling Strategy Our rejection sampling approach is effective for inferring thinking traces when the generator model is capable enough to overlap with human judgment. However, its efficacy may be reduced in tasks with large alignment gaps. The model might struggle to generate a trace that matches the human's

final label, even when multiple candidate responses are sampled. To mitigate this, future research could explore sampling strategies that enable a broader coverage of possible thinking traces in the space. In addition, our current method infers thinking traces for an aggregated rating from multiple raters. Future work could instead infer thinking traces for each individual annotator, which may enrich the diversity of collected reasoning patterns and provide deeper insights into subjective variation across raters.

Relation with Automated Prompt Optimization While beyond the scope of this paper, our codebook refinement process can also be viewed as a complementary approach to Automated Prompt Optimization (APO) techniques (Pryzant et al., 2023). Many APO methods iteratively refine the prompts, starting from a simple seed that is hand-crafted (Ramnath et al., 2025). Our method, in contrast, provides a semantically rich and data-driven starting point derived from inferred human thinking traces. This refined codebook could serve as a high-quality initial prompt, which can then be passed to established APO algorithms for further fine-tuning. This two-stage approach could bridge the gap between human-centric guideline design and automated optimization, leading to prompts that are both effective and grounded in human cognitive patterns.

Involving Human Annotators Another promising direction is to create a lightweight form of human-AI collaboration for future data annotation. This approach achieves a practical tradeoff between the high complexity of asking humans to write thinking traces from scratch and the need for an accurate thinking trace. Future workflows could first use a model to generate several candidate thinking traces given a human rating. A human rater would then simply select the trace that best reflects their own reasoning process. This "select-and-validate" method is significantly less demanding for annotators, but still yields the high-fidelity data needed to further strengthen the reliability of downstream LLM raters. Furthermore, inferred thinking traces have immediate applications for human-centered tasks. They can serve as valuable training resources to accelerate the onboarding of new human annotators, and the refined annotation codebooks could be used not only to guide LLMs but also to improve inter-rater reliability among human teams – a hypothesis that we can test in future work.

6 Conclusion

Our work introduces a scalable framework to infer the latent thinking traces of human annotators using reasoning language models. We demonstrate that these inferred traces serve as high-fidelity proxies for the reasoning paths of human judges across two complementary case studies. First, their integration into reasoning-enhanced fine-tuning significantly improves the alignment of open-weight LLM raters to human ratings. Second, these traces can be used to automatically refine annotation guidelines, which in turn boosts agreement among multiple LLM raters, as well as their agreement with human judgments. Our findings highlight the potential to unlock vast, label-only datasets, transforming them into transparent, reasoning-rich resources for building more reliable and interpretable LLM-based evaluators.

References

Maryam Amirizaniani, Elias Martin, Maryna Sivachenko, Afra Mashhadi, and Chirag Shah. 2024. Do LLMs exhibit human-like reasoning? Evaluating theory of Mind in LLMs for open-ended responses. *arXiv [cs.CL]* (June 2024).

Ron Artstein and Massimo Poesio. 2008. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics* 34, 4 (2008), 555–596.

Rewina Bedemariam, Natalie Perez, Sreyoshi Bhaduri, Satya Kapoor, Alex Gil, Elizabeth Conjar, Ikkei Itoku, David Theil, Aman Chadha, and Naumaan Nayyar. 2025. Potential and perils of large language models as judges of unstructured textual data. *arXiv preprint arXiv:2501.08167* (2025).

Adrien Bibal, Nathaniel Gerlek, Goran Muric, Elizabeth Boschee, Steven Fincke, Mike Ross, and Steven N Minton. 2025. Automating annotation guideline improvements using LLMs: A case study. (2025), 129–144.

Samuel Carton, Anirudh Rathore, and Chenhao Tan. 2020. Evaluating and Characterizing Human Rationales. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 9294–9307. doi:10.18653/v1/2020.emnlp-main.747

Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA.

Cyril Chhun, Pierre Colombo, Chloé Clavel, and Fabian M Suchanek. 2022. Of human criteria and automatic metrics: A benchmark of the evaluation of story generation. *arXiv preprint arXiv:2208.11646* (2022).

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261* (2025).

Scott Crossley, Perpetual Baffour, Jules King, Lauryn Burleigh, Walter Reade, and Maggie Demkin. 2024. Learning Agency Lab - Automated Essay Scoring 2.0. `https://kaggle.com/competitions/learning-agency-lab-automated-essay-scoring-2`. Kaggle.

Ishita Dasgupta, Andrew K. Lampinen, Stephanie C. Y. Chan, Hannah R. Sheahan, Antonia Creswell, Dharshan Kumaran, James L. McClelland, and Felix Hill. 2024. Language models show human-like content effects on reasoning tasks. arXiv:2207.07051 [cs.CL] `https://arxiv.org/abs/2207.07051`

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. ERASER: A benchmark to evaluate rationalized NLP models. *arXiv preprint arXiv:1911.03429* (2019).

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. RAFT: Reward rAnked FineTuning for generative foundation model alignment. *arXiv [cs.LG]* (April 2023).

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2020. SummEval: Re-evaluating Summarization Evaluation. *arXiv preprint arXiv:2007.12626* (2020).

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

Carlos Gómez-Rodríguez and Paul Williams. 2023. A confederacy of models: A comprehensive evaluation of LLMs on creative writing. *arXiv preprint arXiv:2310.08433* (2023).

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024a. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594* (2024).

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2024b. A Survey on LLM-as-a-Judge. *arXiv [cs.CL]* (Nov. 2024).

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

Andrew Halterman and Katherine A Keith. 2025. Codebook LLMs: Evaluating LLMs as measurement tools for political science concepts. *arXiv [cs.CL]* (Jan. 2025).

Enshuo Hsu, Martin Ugbala, Krishna Kumar Kookal, Zouaidi Kawtar, Nicholas L Rider, Muhammad F Walji, and Kirk Roberts. 2025. Synthesized annotation guidelines are knowledge-lite boosters for clinical information extraction. *arXiv [cs.CL]* (April 2025).

Mete Ismayilzada, Claire Stevenson, and Lonneke van der Plas. 2024. Evaluating creative short story generation in humans and large language models. *arXiv preprint arXiv:2411.02316* (2024).

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720* (2024).

V K. Chaithanya Manam, Dwarakanath Jampani, Mariam Zaim, Meng-Han Wu, and Alexander J. Quinn. 2019. TaskMate: A mechanism to improve the quality of instructions in crowdsourcing. In *Companion Proceedings of The 2019 World Wide Web Conference*. ACM, New York, NY, USA.

Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. 2023. Analyzing dataset annotation quality management in the wild. *arXiv [cs.CL]* (July 2023).

Terry K Koo and Mae Y Li. 2016. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of chiropractic medicine* 15, 2 (2016), 155–163.

Michael Krumdick, Charles Lovering, Varshini Reddy, Seth Ebner, and Chris Tanner. 2025. No free labels: Limitations of LLM-as-a-judge without human grounding. *arXiv [cs.CL]* (March 2025).

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using GPT-4 with better human alignment. *arXiv [cs.CL]* (March 2023).

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2024. Aligning with human judgement: The role of pairwise preference in large language model evaluators. *arXiv [cs.CL]* (March 2024).

V K Manam and Alexander Quinn. 2018. WingIt: Efficient refinement of unclear task instructions. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 6 (June 2018), 108–116.

Han Meng, Yitian Yang, Yunan Li, Jungup Lee, and Yi-Chieh Lee. 2024. Exploring the potential of human-LLM synergy in advancing qualitative analysis: A case study on mental-illness stigma. *arXiv [cs.HC]* (May 2024).

Philipp Mondorf and Barbara Plank. 2024. Comparing inferential strategies of humans and large language models in deductive reasoning. *arXiv [cs.CL]* (Feb. 2024).

Cónal O'Connor and Hélène Joffe. 2020. Intercoder Reliability in Qualitative Research: Debates and Practical Guidelines. *Social and Personality Psychology Compass* 14, 2 (2020), e12516.

Vivek Krishna Pradhan, Mike Schaekermann, and Matthew Lease. 2022. In search of ambiguity: A three-stage workflow design to clarify annotation guidelines for crowd workers. *Front. Artif. Intell.* 5 (May 2022), 828187.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495* (2023).

Kiran Ramnath, Kang Zhou, Sheng Guan, Soumya Smruti Mishra, Xuan Qi, Zhengyuan Shen, Shuai Wang, Sangmin Woo, Sullam Jeoung, Yawei Wang, et al. 2025. A systematic survey of automatic prompt optimization techniques. *arXiv preprint arXiv:2502.16923* (2025).

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L Bileschi, Noah Constant, Roman Novak, Rosanne

Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. 2023. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv [cs.LG]* (Dec. 2023).

Saurabh Srivastava, Sweta Pati, and Ziyu Yao. 2025. Instruction-tuning LLMs for event extraction with annotation guidelines. *arXiv [cs.CL]* (Feb. 2025).

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges. *arXiv [cs.CL]* (Aug. 2025).

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. DART-math: Difficulty-Aware Rejection Tuning for mathematical problem-solving. *arXiv [cs.CL]* (June 2024).

Maya Grace Torii, Takahito Murakami, and Yoichi Ochiai. 2024. Expanding horizons in HCI research through LLM-driven qualitative analysis. *arXiv [cs.HC]* (Jan. 2024).

Alicia Y Tsai, Adam Kraft, Long Jin, Chenwei Cai, Anahita Hosseini, Taibai Xu, Zemin Zhang, Lichan Hong, Ed H Chi, and Xinyang Yi. 2024. Leveraging LLM reasoning enhances personalized recommender systems. *arXiv [cs.IR]* (July 2024).

Somin Wadhwa, Silvio Amir, and Byron C Wallace. 2024. Investigating mysteries of CoT-augmented distillation. *arXiv [cs.CL]* (June 2024).

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024. Helpsteer 2: Open-source dataset for training top-performing reward models. *Advances in Neural Information Processing Systems* 37 (2024), 1474–1501.

Joel P Wiebe, Rubaina Khan, Samantha Burns, and James D Slotta. 2025. Qualitative research in the age of LLMs: A human-in-the-loop approach to hybrid thematic analysis. In *Proceedings of the International Conference of the Learning Sciences*. International Society of the Learning Sciences, 1123–1131.

Ziang Xiao, Xingdi Yuan, Q Vera Liao, Rania Abdelghani, and Pierre-Yves Oudeyer. 2023. Supporting qualitative analysis with large language models: Combining codebook with GPT-3 for deductive coding. *arXiv [cs.CL]* (April 2023).

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D Goodman. 2022. STaR: Bootstrapping reasoning with reasoning. *arXiv [cs.LG]* (March 2022), 15476–15488.

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. *Advances in neural information processing systems* 36 (2023), 46595–46623.

Xin Zhou, Kisub Kim, Ting Zhang, Martin Weyssow, Luis F Gomes, Guang Yang, and David Lo. 2025. An LLM-as-Judge Metric for Bridging the Gap with Human Evaluation in SE Tasks. *arXiv preprint arXiv:2505.20854* (2025).

A Dataset Details

HANNA The HANNA dataset (Human-ANnotated Narratives for ASG evaluation)(Chhun et al., 2022) is a benchmark of 1056 stories generated from 96 prompts. Each story was annotated by 3 native English speakers from Amazon Mechanical Turk with Masters Qualifications along six defined evaluation criteria-engagement, complexity, surprise, relevance, coherence, and empathy, resulting in 19008 annotations. Among these, 10245 annotations exhibit a standard deviation lower than 1.0 across raters, which we regard as effective and reliable samples.

Essay The Essay dataset consists of student essays from a Kaggle competition on automated essay scoring(Crossley et al., 2024). The 17307 student essays were annotated with a holistic quality score by expert raters. We regard all of them as effective samples.

SummEval The SummEval dataset is a large-scale benchmark designed to evaluate summarization models(Fabbri et al., 2020). The summaries were generated by 23 summarization models trained on CNN/DailyMail news dataset. Each summary is evaluated by crowd-sourced annotators from Amazon Mechanical Turk and three expert annotators, where two had written academic papers on summarization for conferences, and one had completed a senior thesis on the topic. For our analysis, we focus on the expert annotations and retain only the effective samples, defined as those with a standard deviation below 1.0 across expert ratings. This filtering yields 5645 reliable samples out of 6400 total.

WMT-2020 The WMT20 Metrics Shared Task dataset is a benchmark for evaluating machine translation systems(Freitag et al., 2021). It contains system outputs across multiple language pairs (e.g., Chinese to English, English to German). Each translation was evaluated on both MQM (Multidimensional Quality Metrics) and SQM (Scalar Quality Metrics) by human annotators. In this work, we focus on the Chinese–English language pair and use the SQM (Scalar Quality Metric) annotations, where each translation is assigned three professional translators native in the target language of overall quality. The dataset contains 19,950 translation pairs, of which 6,934 exhibit a standard deviation below 1.0 across raters. We regard these as reliable samples and use them for training and evaluation.

HelpSteer2 The HelpSteer2 dataset is a preference dataset for training reward models that align large language models with human preferences(Wang et al., 2024). It contains 10,681 prompts, each paired with two responses, and every response is annotated on five dimensions: correctness, helpfulness, coherence, complexity, and verbosity. To ensure annotation reliability, each response is initially rated by three annotators, with two additional annotations collected when the disagreement among the first three annotations is high. In total, the dataset comprises 21,362 annotated responses, all of which we treat as effective samples. For our experiments, we focus on the dimensions of correctness and helpfulness.

B SFT Details

We trained all models using LoRA and chose AdamW as the optimizer. The hyperparameters are: learning rate = 5e-5 (warmup = 100 steps), batch size = 128, weight decay = 0.01, LoRA $\alpha = 32$, LoRA r = 16.

C Prompts

Instruction refinement

<instruction>
You will be given a list of analysis on scoring the `{dimension}` of `{generation_type}`. Your task is to extract a concrete and concise step-by-step instruction from the analysis that could be easily followed by an annotator without any training. Based on your extraction, improve the original annotation guidelines (<original_codebook>...</original_codebook>). Only improve the instruction part (e.g. thinking path to follow), do not change the rubric part (e.g. criteria). Your final annotation guidelines should be put into <codebook>...</codebook> tag.
</instruction>

<original_codebook>
`{original_codebook}`
</original_codebook>

<analysis>
`{analysis}`
</analysis>

Critique extraction from CoT

<instruction>
You will be given an analysis on scoring the `{dimension}` of a `{generation_type}`. Your task is to list all the critique from the analysis that help evaluate the `{dimension}` of the `{generation_type}`. Your response should be a list of sentences, each on a new line, without bullet points. Your response should not include any other text.
</instruction>

<analysis>
`{cot}`
</analysis>

Rubric refinement

<instruction>
You will be shown a list of diverse critiques on the `{dimension}` of `{generation_type}`. Each of the critique is corresponding to a specific rating level. Your task is to summarize the critiques into a single rubric, based on the pattern of the different rating levels. Based on the new criteria, improve the original annotation guidelines. Only modify the criterion part. Your final annotation guidelines should be put into <codebook>...</codebook> tag.
</instruction>

<original_codebook>
`{original_codebook}`
</original_codebook>

<critiques>
`{critiques}`
</critiques>

D Complete Examples of Annotation Codebooks

A complete list of codebooks before and after refinement is provided in our anonymous codebase: `https://anonymous.4open.science/r/thru_judge_eye-56DD/codebooks/`. Here we show a complete refined version of Figure 4.

Read the Materials Thoroughly Start by reading the prompt, the human story, and the subject story. Note that the subject story is the one being rated, not the human story. First, read the prompt to understand the context. Read the Human Story to get a sense of a complete narrative for comparison, if relevant. Read the Subject Story thoroughly, as this is the story you need to evaluate.

Step-by-Step Rating:

- Look for key elements such as characters, events, plot, themes, or setting ...
- Evaluate if the identified elements are developed, interconnected, and coherent ...
- Consider whether the story follows a structured progression (linear or non-linear) ...
- Stories with minimal elements or undeveloped content are simpler. Stories that attempt multi-layered exploration of ideas, even if not fully realized, should be evaluated as more complex.

Select the Rating

1. The story is very simple, lacks depth, and introduces no intricate or developed elements.

2. The story is somewhat simple, with few elements that are underdeveloped, fragmented, or disconnected.

3. The story demonstrates some complexity, introducing multiple elements, but lacks depth, development, or coherence in integrating these elements.

4. The story is mostly complex, weaving together several developed elements with minor areas that could be expanded or refined.

5. The story is highly elaborate and complex, integrating multiple layers (e.g., detailed world-building, character depth, advanced structure, thematic sophistication) in a cohesive and interconnected way.

Consider overall structure and presentation

- Are there abrupt shifts, confusing elements, or fragmented storytelling that make it harder to engage with the story?
- Does the story have vivid details or emotional resonance that enhance engagement despite structural issues?

Handle specific cases

- If the story is cut off mid-sentence, rate it as if it ended just before the unfinished sentence.
- If the story is not relevant to the prompt, focus only on how engaging it is—irrelevance affects the Relevance criterion, not Engagement.

Keep your focus

- Rate solely based on the Engagement criterion, even if the story lacks relevance to the prompt.
- Do not let your assessment be influenced by the Human Story except as a potential comparison of narrative structure or clarity.