RETRACEQA: Evaluating Reasoning Traces of Small Language Models in Commonsense Question Answering

Francesco Maria Molfese*, **Luca Moroni***, **Ciro Porcaro, Simone Conia, Roberto Navigli**
Sapienza NLP Group, Sapienza University of Rome
{molfese, moroni, porcaro, conia, navigli}@diag.uniroma1.it

Abstract

While Small Language Models (SLMs) have demonstrated promising performance on an increasingly wide array of commonsense reasoning benchmarks, current evaluation practices rely almost exclusively on the accuracy of their final answers, neglecting the validity of the reasoning processes that lead to those answers. To address this issue, we present RETRACEQA, a novel benchmark that introduces process-level evaluation for commonsense reasoning tasks. Our expert-annotated dataset reveals that in a substantial portion of instances (14-24%), SLMs provide correct final answers despite flawed reasoning processes, suggesting that the capabilities of SLMs are often overestimated by evaluation metrics that focus only on comparing the final answer with the ground truth. Indeed, we show that, when employing strong Large Language Models (LLMs) as automated judges for reasoning-aware evaluation rather than answer-only metrics, SLM performance drops significantly across all models and datasets, with scores decreasing by up to 25%.

1 Introduction

Recent work in language modeling has led to effective SLMs with impressive performance levels across various benchmarks (Grattafiori et al., 2024; Abdin et al., 2024; Fourrier et al., 2024; Qwen et al., 2025). However, current evaluation practices rely almost exclusively on final answer accuracy, i.e., counting an instance as correct when the model's prediction matches the ground truth, regardless of the reasoning process. This answer-centric approach overlooks a fundamental factor: models can arrive at correct answers through invalid reasoning paths, artificially inflating performance metrics and masking important weaknesses in their actual reasoning capabilities.

To this end, the research community has recently proposed several benchmarks to examine "reasoning traces" – the step-by-step explanations generated by language models to arrive at their final answers – in a more systematic way (Zheng et al., 2024; Zeng et al., 2024a,b; Tyen et al., 2024). These benchmarks are necessary for the development and evaluation of automatic approaches, such as Process Reward Models (PRMs) (Lightman et al., 2023; Wang et al., 2024b; Zhang et al., 2025) and LLMs as judges (Gu et al., 2025), aimed at identifying the specific location of errors within reasoning traces and not just the correctness of the final answer (Zheng et al., 2024; Zeng et al., 2024a,b; Tyen et al., 2024).

However, contemporary work faces two key limitations. First, existing benchmarks focus primarily on mathematics and science, leaving reasoning processes in other areas like commonsense reasoning largely underexplored, despite requiring fundamentally different capabilities. Second, specialized PRMs and LLMs employed as judges are typically used to optimize task performance through feedback during fine-tuning or Best-of-N sampling, rather than evaluating whether reasoning traces that reach correct answers contain intermediate errors, potentially leading to inflated performance assessments. Therefore, our research question is: *how can we effectively evaluate reasoning processes in commonsense, and to what extent do current answer-only metrics misrepresent SLM capabilities?*

To address these limitations, we provide the following contributions:

- We introduce RETRACEQA, the first benchmark for evaluating reasoning traces of SLMs in commonsense reasoning tasks, including a set of 2,421 reasoning traces manually annotated with step-level error locations and qualitative error categorizations;
- Quantitative evidence that up to 24% of flawed reasoning traces still produce the cor-

* Equal contribution.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 38817–38832

rect final answer, demonstrating how current answer-only evaluations significantly overestimate model capabilities;

- Comprehensive reference-based evaluation of both closed and open-source LLMs as judges, revealing that, while models can often detect whether a trace is correct as a whole, they struggle to identify the exact location of reasoning errors;
- Reference-free evaluation of LLM-as-a-judge models and mathematical PRMs applied to commonsense reasoning, revealing substantial performance degradation when transferring across domains.

Our findings show that answer-only metrics substantially overestimate SLM performance, with scores dropping by up to 25% when accounting for reasoning correctness, also highlighting the need for reasoning-aware evaluation beyond STEM domains. RETRACEQA provides both a practical benchmark and strong evidence that current evaluation practices can misrepresent reasoning in commonsense question answering tasks. We release our benchmark, code, and data at https://github.com/SapienzaNLP/ReTraceQA.

2 Related Work

Process-Based Evaluation Approaches. The research community has introduced two main approaches for assessing reasoning quality beyond final answers: Process Reward Models (PRMs) and LLM-as-a-judge. PRMs are specialized models fine-tuned to evaluate the correctness of reasoning steps, in contrast to Outcome Reward Models (ORMs), which focus solely on final answers (Uesato et al., 2022; Lightman et al., 2023). PRMs specifically aim to identify the first erroneous step in a reasoning trace, enabling both targeted feedback for model training and quality filtering in Best-of-N selection scenarios (Pan et al., 2023; Wang et al., 2024b). PRMs can be built and trained in several ways: Lightman et al. (2023) used human-labeled data for error detection, while Li et al. (2023) and Wang et al. (2024b) employed Monte Carlo estimation to determine the probability of a chain of steps to be correct. More recent work by Hosseini et al. (2024) and Zhang et al. (2025) leverages larger LLMs as automated judges to generate training signals for PRMs, creating a teacher-student paradigm for reasoning evaluation.

In parallel to specialized PRMs, general-purpose LLMs prompted as judges have emerged as an effective — albeit expensive — alternative approach. These models assess reasoning trace validity without task-specific training, providing both binary correctness judgments and localized error identification (Zheng et al., 2024). While more flexible than PRMs, judges may lack the specialization that targeted training provides.

Benchmarks for Reasoning Evaluation. Several benchmarks have been developed to evaluate models' abilities to identify errors in reasoning traces, each with distinct characteristics. ProcessBench (Zheng et al., 2024) specifically targets reasoning error identification by requiring models to indicate the exact location of incorrect steps within mathematical reasoning traces. MR-Ben and MR-GSM8K (Zeng et al., 2024a,b) offer more comprehensive meta-reasoning assessment, including error localization, error explanation, and suggested corrections. Findings from these benchmarks consistently demonstrate that even state-of-the-art LLMs struggle to detect reasoning error locations accurately, though they do show potential for providing helpful corrections once errors have been explicitly identified (Tyen et al., 2024; Huang et al., 2024).

Limitations and Research Gaps. Despite progress in process-based evaluation, existing work presents three key limitations. First, PRMs and judge models tend to be used primarily for Best-of-N selection, thus for ranking multiple outputs in order to improve generation rather than as tools for validating reasoning traces during evaluation. Second, current benchmarks are largely restricted to mathematical domains, overlooking reasoning types found in commonsense tasks that involve qualitatively different inference. Third, the implications of reasoning-aware evaluation on SLM assessment remain underexplored, particularly in respect of how final answer metrics can misrepresent underlying reasoning quality.

3 ReTraceQA

In this section, we introduce RETRACEQA, our novel gold benchmark designed to assess the ability of LLMs to determine whether a reasoning trace of an SLM is correct, or if it is not, to identify the specific step where an error occurs. In the following, Section 3.1 provides the formal task definition, Section 3.2 describes the datasets selected for

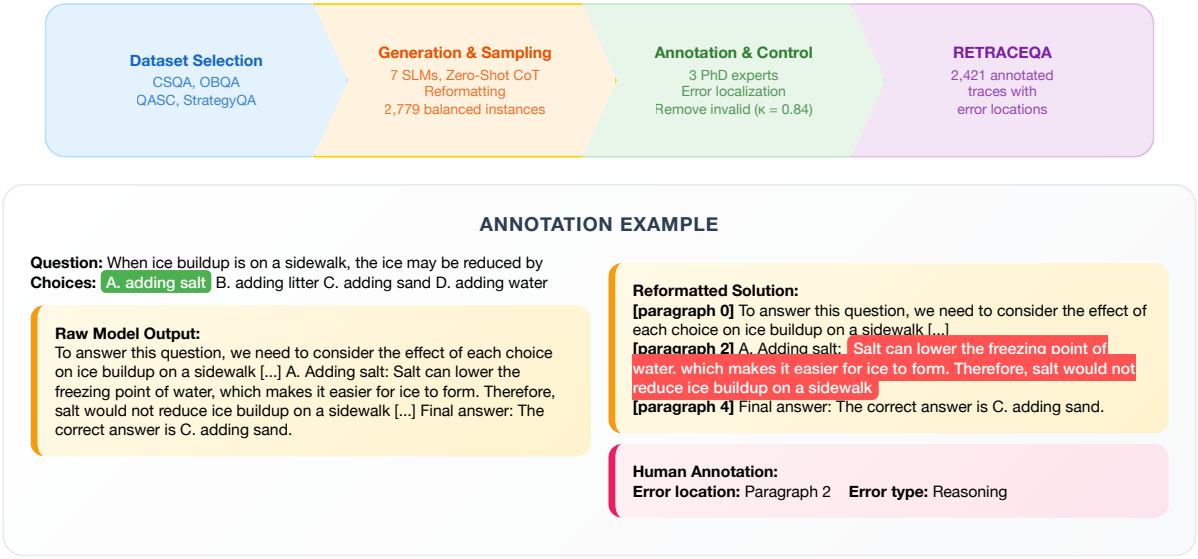


Figure 1: Overview of the RETRACEQA pipeline (top) and a representative example of a fully-annotated instance (bottom). The bottom panel illustrates the identification of error location and categorization, tracing the transformation from the initial raw model output to the final reformatted and structured reasoning trace.

the benchmark, Section 3.3 explains how reasoning traces are generated using a range of SLMs, Section 3.4 explains how we divide the reasoning traces into discrete steps, Section 3.5 details the human annotation process used to construct the resulting resource, and, finally, Section 3.6 presents descriptive statistics of our benchmark. Figure 1 shows the overview of the whole pipeline together with an example of annotation.

3.1 Task Definition

Given a commonsense reasoning question, the goal is to evaluate the validity of an SLM-generated reasoning trace by identifying the earliest step at which an error occurs, if any. Formally, let q denote the input question (with an optional set of choices C), and let $S = [s_0, s_1, \ldots, s_n]$ represent the step-by-step reasoning trace. The task is to predict an index $i \in \{-1, 0, \ldots, n\}$, where $i = -1$ signifies that all reasoning steps are correct, and $i \geq 0$ indicates that the first error occurs at step s_i. This formulation is in line with state-of-the-art benchmarks like ProcessBench (Zheng et al., 2024), in which the authors note that, for steps after the first error, the meaning of their correctness may become ambiguous or debatable. Indeed, derivations based on incorrect premises can make sense, but still remain on a globally incorrect reasoning path (Lightman et al., 2023). Based on this assumption, we choose to focus on identifying the earliest-occurring error in the reasoning traces.

3.2 Dataset Selection

To construct our benchmark, we source questions from four widely-used datasets in commonsense reasoning: CommonsenseQA (Talmor et al., 2019, CSQA), OpenBookQA (Mihaylov et al., 2018, OBQA), QASC (Khot et al., 2020), and StrategyQA (Geva et al., 2021), all of which are multiple-choice or binary question answering datasets that provide a question along with a set of candidate answers. These datasets primarily target commonsense reasoning grounded in general world knowledge, but also feature questions involving encyclopedic and subject-specific knowledge, as well as reasoning over spatial, temporal, or causal relationships. When test set labels are not publicly available, we follow standard practice and instead utilize the corresponding development sets (Liu et al., 2023; Molfese et al., 2024); this applies to CSQA and QASC. Together, these datasets span a range of reasoning challenges across commonsense domains, making them well-suited for evaluating the correctness and robustness of reasoning traces.

3.3 Solution Generation

For each instance in our selected datasets, we generate step-by-step reasoning traces using SLMs from the widely used LLaMA, Phi and Qwen families of open-source language models (Grattafiori et al., 2024; Abdin et al., 2024; Qwen et al., 2025). We follow standard practice and define an SLM as any language model with no more than 10 billion

parameters (Fu et al., 2023; Wang et al., 2024a). Specifically, we use the following instruction-tuned variants: Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, Llama-3.2-1B-Instruct, Llama-3.2-3B-Instruct, Llama-3.1-8B-Instruct, and Phi-4-mini-instruct. This selection enables us to examine performance variation within model families as model size increases (with the exception of Phi, which is only available in a single size under 10 billion parameters), while also capturing differences across architectures. We generate traces by prompting models with a zero-shot Chain-of-Thought (CoT) setup (Wei et al., 2023; Kojima et al., 2023), which encourages step-by-step reasoning without conditioning on specific examples.

Initially, we collect a total of 3,334 original questions distributed across the datasets as follows: 1,221 questions from CSQA, 500 questions from OBQA, 926 questions from QASC and 687 questions from StrategyQA. For each original question, we generate reasoning traces using 7 distinct SLMs, resulting in an initial pool of 23,338 total reasoning traces. Then, we perform careful sampling from this initial pool to ensure three factors simultaneously: (i) balanced representation of correct and incorrect traces in terms of final answer accuracy, (ii) balanced representation of each model and (iii), uniqueness of each question. This sampling step reduces the dataset to a total of 2,779 unique instances (i.e., each instance is a unique question associated with exactly one reasoning trace). Details on reasoning trace generation and answer-based classification can be found in Appendices A and B, respectively.

3.4 Solution Reformatting

A key step in building RETRACEQA involves ensuring that model-generated reasoning traces are segmented into coherent, interpretable steps. In mathematical domains, prior work has shown that automatic solutions often require post-hoc re-segmentation due to formatting inconsistencies and unclear boundaries between reasoning steps (Zheng et al., 2024). In contrast, we find that, for commonsense reasoning tasks, step segmentation emerges more naturally. When using CoT prompting, models tend to produce clearly delineated reasoning traces, with each step expressed as a self-contained sentence, making reformatting unnecessary.[1]

[1]We split reasoning traces at "\n\n", falling back to "\n" only when the double-newline sequence is not found.

3.5 Human Annotation

To construct a benchmark that enables both binary reasoning evaluation and fine-grained error localization, we annotate a diverse set of SLM-generated reasoning traces with step-level error information. The annotation task follows the setup described in Section 3.1: for each reasoning trace, annotators are asked to identify the earliest step that contains an error, or to indicate that the entire trace is correct. Additionally, annotators are asked to assign one of three available labels classifying the nature of the error. To ensure high-quality reproducible labels and to minimize the inherent subjectivity in evaluating commonsense, we employ a hierarchical error taxonomy. We categorize errors from grounding to inference, ensuring that each error is assigned to a single, mutually exclusive category based on its primary cause. Specifically, a step is considered erroneous if it falls into one of the following categories:

- **Misinterpretation (Grounding Level):** errors occurring when the model misunderstands the question, choice meanings, or task requirements. This includes misrepresenting previous steps, referencing non-existent choices, or providing multiple answers.
- **Hallucination (Content Level):** errors involving the introduction of empirically false or unverifiable world knowledge. This category is strictly reserved for instances where the model's logic might be structurally sound, but the "factual" building blocks are incorrect (e.g., stating "wolves are not found in arctic regions"). A step is labeled as a hallucination only if the input was correctly interpreted, but the world knowledge used by the model is false or unverifiable.
- **Reasoning (Inference Level):** errors involving the logical transitions between or within steps. This occurs when the model connects correct premises or facts using an invalid logical leap. For example, a model might correctly state that "salt lowers the freezing point of water" but incorrectly infer that "this makes it easier for ice to form." This label is applied when the factual content of the step is correct, but the deduction is logically unsound.

Three expert annotators with PhD-level backgrounds in computer science or linguistics – who

Input	Reasoning Trace	Error Category
Question: Hand dryers can also be used to **Choices**: A. keep cold drinks cool B. dry out wet clothes C. hydrate your face and hands D. make a damp rag damper	**Paragraph 0:** To answer this question, I'll analyze each option and eliminate the incorrect ones based on the properties and common uses of hand dryers. [...] **Paragraph 5:** I will choose option E, which is not provided in the list. Since the correct answer is not among the options, I will choose the one that is least likely to be correct.	Misinterpretation.
Question: Where can bees find the materials needed to make honey? **Choices**: A. heated by the sun B. occurs over a wide range C. trees and flowers D. buildings E. hives F. leaves G. 70-75 degrees Fahrenheit H. interior chambers	**Paragraph 0:** To answer this question, I need to consider the characteristics of honey and its production process. Honey is produced by honeybees (Apis mellifera) through a process called nectar collection and regurgitation [...] **Paragraph 5:** F. leaves: Leaves are the primary source of nectar for honey production.	Hallucination.
Question: During the winter hunt he could hear every motion in the woods, this was because of the what of everything? **Choices**: A. being still B. silence C. stationary D. stillness E. standing still	**Paragraph 0:** The question states that the person can hear every motion in the woods, which implies that there is something that allows them to be aware of even the slightest movements. [...] **Paragraph 3:** The word "being still" (A) implies that the person is not moving, which would make it harder to hear movements, not easier.	Reasoning.

Table 1: Examples of the three error categories identified in RETRACEQA. For each instance, we present the input question and choices, the segmented reasoning trace generated by the SLM, and the final error categorization.

are also the authors of this paper – perform the annotation. Each annotator is given the SLM-generated reasoning trace, the original question, optional answer choices and supporting facts, and the gold answer from the dataset, and is instructed to judge correctness based solely on the reasoning trace, not on the final answer alone. Annotators are also asked to flag problematic instances using a dedicated INVALID tag. These include: ambiguous questions with multiple plausible answers, grammatical or structural issues that impair interpretation and labeling inconsistencies in the original dataset (e.g., an incorrect gold answer). To safeguard the quality of the final benchmark, we exclude all flagged instances. This results in a total of 2,421 clean and fully annotated examples in RETRACEQA.

To evaluate annotation consistency, we randomly sample 25 instances from each of the four datasets in the benchmark (100 total). All three annotators independently label this subset following the same guidelines. Inter-annotator agreement, measured via Fleiss's kappa, yields a score of 0.84, indicating an *"almost perfect"* agreement according to standard interpretation (Landis and Koch, 1977).

Figure 1 illustrates the transition from raw model output to a fully annotated instance, depicting the reformatted solution alongside its error location and categorization. To further clarify our taxonomy, Table 1 provides representative examples for each of the three error categories, with an additional set of cases detailed in Appendix C. A comprehensive and detailed description of the annotation protocol is available in Appendix D.

3.6 Benchmark Statistics

Table 2 presents a detailed analysis of the reasoning traces across each subset, including the number of samples that reach a correct or incorrect final answer, the proportion of process errors (instances with correct answers but flawed reasoning), descriptive statistics on reasoning trace length and the percentage of instances falling in each of the available error categories.

A key observation is that a non-trivial percentage of responses, averaging to 17.9% across datasets,

	CSQA	OBQA	QASC	StrategyQA
Final samples (error)	296	184	219	271
Final samples (correct)	603	244	245	359
Total samples	**899**	**428**	**464**	**630**
Process errors (%)	16.3	14.7	16.6	24.0
Invalid instances	238	20	46	54
Avg. steps (error)	8.2	8.1	8.0	6.9
Avg. steps (correct)	8.2	7.8	7.9	6.8
Error Category Distribution (%)				
Hallucination	41.9	46.7	47.5	62.5
Reasoning	34.0	34.7	35.4	27.9
Misinterpretation	24.1	18.6	17.1	9.6

Table 2: ReTraceQA statistics. Process errors refer to instances with correct final answers but flawed reasoning. Invalid instances were flagged during annotation and excluded. Error category distributions are calculated over all erroneous traces.

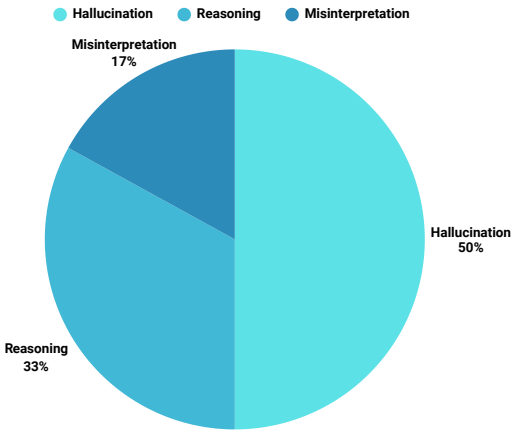


Figure 2: Error category distribution averaged across the four subsets of our ReTraceQA benchmark.

arrive at the correct final answer despite containing a reasoning error. This pattern is consistent with findings from mathematical reasoning benchmarks (Zheng et al., 2024), in which even strong language models are able to reach the correct answer while making mathematical mistakes, and highlights a critical limitation of standard evaluation practices, which often overlook flawed intermediate reasoning when only final answers are assessed. As a result of such oversight, leaderboard metrics may overestimate the true reasoning capability of language models. Moreover, we can see a consistent distribution of error categories across the four subsets of our benchmark. Specifically, hallucination errors constitute the majority of failures (41.9%–62.5%), followed by reasoning errors (27.9%–35.4%) and misinterpretation errors (9.6%–24.1%). In Figure 2 we report the error category distribution averaged across the four subsets of our benchmark. This suggests that SLMs struggle primarily with factual grounding, frequently generating unverifiable claims or incorrect assumptions, though logical coherence issues also remain prevalent, accounting for roughly one-third of all errors. The lower proportion of misinterpretation errors indicates that models generally understand task requirements and question semantics, but fail both in anchoring their reasoning in accurate world knowledge and in maintaining sound logical inference chains. Individual statistics for each model are provided in Appendix E.

4 Experimental Setup

Our benchmark evaluates LLMs along two axes: (1) reference-free assessment of SLM reasoning trace validity in order to determine whether models can reliably provide fine-tuning feedback or perform Best-of-N selection without ground truth labels, and (2) reference-based assessment where models judge reasoning traces using both the correct answer and reasoning process, extending evaluation beyond final answer correctness alone. In the following, we list the models used for our experiments (Section 4.1) and the evaluation metrics for both reference-free and reference-based settings (Section 4.2).

4.1 Models

LLM-as-a-judge. We follow recent work on automated evaluation (Zheng et al., 2023) by prompting LLMs to assess SLM reasoning traces. The prompt is slightly adapted from prior work (Zheng et al., 2024) to better suit commonsense reasoning tasks (Appendix F). We evaluate the following set of open-weight and closed models: Mistral-Small-24B-Instruct-2501 (Mistral, 2025), Llama-3.3-70B-Instruct (Grattafiori et al., 2024), Qwen2.5-72B-Instruct (Qwen et al., 2025), Gemini-2.0-Flash (DeepMind, 2025), DeepSeek-R1 (DeepSeek-AI et al., 2025), GPT-4o-mini (OpenAI et al., 2024a), GPT-4o (OpenAI et al., 2024a) and o1-mini (OpenAI et al., 2024b). Greedy decoding is used for all models except o1-mini and DeepSeek-R1, for which we report performance using a sample at temperature 1.0 due to API constraints.

Process Reward Models. We evaluate several publicly available PRMs by extracting their stepwise correctness predictions and identifying the first step flagged as incorrect. The evaluated models fall into three groups: (1) math-shepherd-

mistral-7B (Wang et al., 2024b), which uses empirical correctness likelihoods over reasoning steps; (2) Skywork-o1-Open-PRM-Qwen-2.5-1.5B and Skywork-o1-Open-PRM-Qwen-2.5-7B (Skywork, 2024), which output raw scalar scores; (3) Qwen2.5-Math-7B-PRM800K and Qwen2.5-PRM-7B (Zheng et al., 2024), fine-tuned respectively on the PRM800K dataset and on synthetic data derived from LLM-as-a-judge annotations.

For models in groups (1) and (3), trained with sigmoid activations over each step, we determine step correctness by rounding predictions to the nearest integer (1 = correct, 0 = incorrect). For models in group (2), we select a threshold that maximizes F1 on a validation split of CSQA, following Zheng et al. (2024), and use it to round scalar scores.

4.2 Evaluation Metrics

Reasoning Trace Evaluation. For both reference-free and reference-based settings, we evaluate models using two complementary metrics: *correct*, measuring accuracy in identifying fully valid traces (human-labeled as -1), and *error*, measuring accuracy in localizing the first erroneous step in flawed traces (human-labeled as i, where $i \geq 0$). These metrics assess whether LLMs can provide targeted feedback to SLMs during training and quantify their reliability for Best-of-N scoring during evaluation. Following prior work (Zheng et al., 2024), we report the harmonic mean (F1) of *correct* and *error* to balance overly permissive versus overly critical model behaviors.

Downstream SLM Evaluation. To assess the impact of reasoning-aware evaluation on SLMs using LLM-as-a-judge, we employ the best-performing judge from the reference-based evaluation on RETRACEQA under two configurations: (1) answer-only evaluation (simulating standard approaches) and (2) full trace validation (accepting predictions only when both reasoning and answers are correct). We measure performance using *accuracy* (correctly distinguishing valid from invalid traces: $i = -1$ vs. $i \neq -1$) and *error recall* (identifying flawed traces where both model and human annotations indicate $i \neq -1$). We evaluate seven SLMs with the same judge under the two settings across four commonsense datasets, generating diverse reasoning traces at temperature 0.7 to ensure variety while avoiding overlap with our annotated benchmark.

5 Results

In this section, we first present LLM performance on reference-free and reference-based evaluation on RETRACEQA (Section 5.1). We then report downstream SLM evaluation results, where the best-performing LLM-as-a-judge on RETRACEQA is used to assess SLM-generated outputs across multiple benchmarks (Section 5.2).

5.1 Reasoning Trace Evaluation

Reference-free Evaluation. Table 3 (top) highlights substantial limitations of LLM-as-a-judge and PRMs when asked to assess SLM reasoning traces in commonsense reasoning tasks under reference-free evaluation. Overall, F1 scores across all four datasets remain relatively low, even for the strongest judges, suggesting that reliably evaluating the soundness of reasoning traces remains a challenging task. While state-of-the-art LLMs like GPT-4o and o1-mini outperform others with F1 scores exceeding 60% on some datasets, the average F1 across models hovers around 54–56%, indicating considerable room for improvement. These results underscore a key challenge in deploying LLM-as-a-judge models in reinforcement learning or Best-of-N selection settings: their current inability to robustly identify and reward correct intermediate reasoning without having access to the correct label limits their usefulness for guiding reasoning-focused learning objectives.

Additionally, PRMs originally developed and trained for mathematical tasks perform significantly worse across all datasets, with average F1 scores often below 25%. This performance gap emphasizes the PRMs' limited generalization capabilities when transferred to commonsense reasoning (see Pisano and Navigli (2026) for extending the logical capabilities of PRMs).

Reference-based Evaluation. Table 3 (bottom) reports reference-based evaluation results across the four subsets in RETRACEQA. Most models achieve moderate to strong performance in identifying globally correct reasoning traces, but accurately localizing specific error steps remains substantially more challenging. Model size correlates positively with performance. For instance, Qwen2.5-72B-Instruct outperforms Mistral-Small-24B-Instruct by +35.5% F1 on average. However, scale alone is insufficient: DeepSeek-R1, despite being larger than Qwen2.5-72B-Instruct, underperforms across all datasets, suggesting that architec-

Model	CSQA			OBQA			QASC			StrategyQA			Avg. F1
	correct	error	F1	correct	error	F1	correct	error	F1	correct	error	F1	
Process Reward Models (reference-free evaluation)													
Math-Shepherd-PRM-7B	95.3	4.2	8.0	92.4	6.1	11.5	70.2	10.2	17.9	58.7	18.7	28.4	16.5
Skywork-PRM-1.5B	**97.5**	1.9	3.7	**95.6**	2.5	4.8	**88.9**	2.7	5.3	93.0	1.9	3.8	4.4
Skywork-PRM-7B	83.0	9.1	16.4	77.6	11.0	19.3	57.3	10.6	17.9	86.4	6.2	11.6	16.3
Qwen2.5-Math-7B-PRM800K	89.6	13.8	23.8	88.5	20.8	33.7	73.7	24.6	36.9	**97.2**	8.9	16.3	27.7
Qwen2.5-Math-PRM-7B	86.8	**20.9**	**33.8**	81.4	**28.9**	**42.8**	70.2	**37.2**	**48.6**	79.8	**24.5**	**37.4**	**40.7**
Average	90.4	10.0	17.1	87.1	13.9	22.4	72.1	17.1	25.3	83.0	12.0	19.5	21.1
LLM-as-a-judge (reference-free evaluation)													
Mistral-Small-24B-Instruct	25.7	48.3	33.6	34.9	53.1	42.2	22.8	50.9	31.5	24.4	43.6	31.3	34.7
LLaMA-3.3-70B-Instruct	**87.2**	33.3	48.2	87.9	44.5	59.1	**83.6**	38.9	53.1	**91.1**	32.6	48.0	52.1
Qwen2.5-72B-Instruct	85.9	44.1	58.3	80.3	53.5	64.2	77.8	43.0	55.4	84.5	45.1	**58.8**	59.2
DeepSeek-R1	56.6	**57.1**	56.8	49.7	**62.5**	55.4	52.6	**55.6**	54.1	44.1	**63.5**	52.1	54.6
Gemini-2.0-Flash	82.9	46.2	59.3	88.5	57.9	**70.1**	77.2	51.2	61.6	87.3	40.8	55.6	61.7
GPT-4o	86.4	47.8	**61.5**	**89.6**	52.2	66.0	79.5	52.9	**63.5**	89.7	39.1	54.4	61.4
GPT-4o-mini	62.5	47.1	53.7	71.0	54.3	61.5	49.1	49.5	49.3	75.1	39.8	52.0	54.1
o1-mini	82.3	46.9	59.7	79.8	61.6	69.5	73.7	54.9	62.9	77.0	45.1	56.9	**62.3**
Average	71.2	46.3	53.9	72.8	54.9	61.0	64.6	49.6	53.9	71.7	43.7	51.1	55.0
LLM-as-a-judge (reference-based evaluation)													
Mistral-Small-24B-Instruct	22.3	54.3	31.7	32.8	62.4	43.0	21.6	59.4	31.7	14.6	55.4	23.1	32.4
LLaMA-3.3-70B-Instruct	87.7	45.2	59.7	**90.7**	52.2	66.3	85.7	54.3	66.5	82.6	56.6	67.2	64.9
Qwen2.5-72B-Instruct	**89.6**	50.6	64.7	85.3	59.2	69.9	**86.6**	58.4	69.7	83.6	56.4	67.3	67.9
DeepSeek-R1	53.8	**61.5**	57.4	49.2	66.1	56.4	49.7	65.9	56.7	37.6	63.6	47.2	54.4
Gemini-2.0-Flash	86.6	52.2	65.2	89.1	64.1	74.5	78.9	60.4	68.4	66.2	59.0	62.4	67.6
GPT-4o	80.9	58.5	**67.9**	88.0	67.8	76.6	70.8	62.1	66.2	72.8	59.2	65.3	69.0
GPT-4o-mini	62.1	54.1	57.8	68.3	61.6	64.8	52.1	60.4	55.9	39.4	56.1	46.3	56.2
o1-mini	82.6	54.6	65.7	84.7	**74.3**	**79.2**	77.2	**71.3**	**74.2**	**84.5**	**72.9**	**78.3**	**74.4**
Average	70.7	53.9	58.7	73.5	63.5	66.3	65.4	61.5	61.2	60.2	59.9	57.1	60.8

Table 3: Detailed performance of PRMs and LLM-as-a-judge models across the four subsets of RETRACEQA. Each triplet reports accuracy on identifying correct reasoning traces (correct), accuracy on pinpointing the exact error location (error), and overall F1 score. The final column reports the average F1 score across all subsets.

Dataset	o1-mini (ext.)		o1-mini (judge)	
	accuracy	err. rec.	accuracy	err. rec.
CSQA	82.2	65.7	81.9	81.1
OBQA	84.8	74.3	90.2	94.3
QASC	83.0	74.1	86.2	91.5
StrategyQA	74.8	62.8	90.0	92.1
Average	81.2	69.2	**87.0**	**89.8**

Table 4: Accuracy and error recall (%) of o1-mini employed as answer extractor (ext.) and as judge on RETRACEQA. Accuracy measures correct trace classification; error recall measures erroneous trace detection.

tural choices and reasoning-oriented training are critical. The strongest judge, o1-mini, achieves 74.4% F1, highlighting the importance of effective reasoning-oriented objectives. Moreover, we can see that a consistent pattern emerges: models detect trace correctness better than localizing errors. For instance, o1-mini achieves 74.3% error classification on OBQA, still lagging behind its correctness detection ability. Figure 3 compares error position distributions between human annotations and o1-mini predictions. Errors most commonly occur at steps 3-4, suggesting that, while early context establishment succeeds, errors emerge during mid-level inference. We can see that o1-mini's predictions mirror human patterns well, particularly on CSQA and QASC. But they show heavier tails, indicating over-assignment of blame to later steps, potentially capturing error consequences rather than origins. These results highlight both the promise and limitations of LLM-as-a-judge systems: while stronger models align well with human evaluations of overall correctness, precisely identifying error origins remains an open challenge.

5.2 Downstream SLM Evaluation

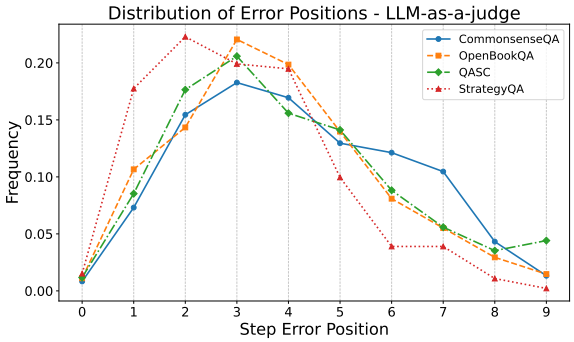
While the reference-based evaluation results in Section 5.1 show that current LLMs employed as automated judges may not reliably localize errors for fine-tuning feedback, their strong performance in assessing overall trace correctness suggests potential for more reliable SLM evaluation. Here we investigate whether reasoning-aware judges that consider both trace validity and final answers provide more accurate assessments than standard answer-only evaluation.

Table 4 demonstrates that the best-performing judge on RETRACEQA (o1-mini) employed as a reasoning-aware judge consistently outperforms

Model *(Instruct)*	CommonsenseQA ext.	CommonsenseQA judge	CommonsenseQA Δ	OpenBookQA ext.	OpenBookQA judge	OpenBookQA Δ	QASC ext.	QASC judge	QASC Δ	StrategyQA ext.	StrategyQA judge	StrategyQA Δ	Avg. Accuracy ext.	Avg. Accuracy judge	Avg. Accuracy Δ
Llama-3.2-1B	47.6	27.7	19.9	47.2	23.8	23.4	47.7	22.2	25.5	53.7	20.1	33.6	49.0	23.4	25.6
Llama-3.2-3B	69.1	58.7	10.4	75.4	58.4	17.0	71.9	51.4	20.5	64.3	34.8	29.5	70.2	50.8	19.4
Llama-3.1-8B	75.7	72.8	2.9	83.0	69.6	13.4	79.4	64.0	15.4	67.2	45.9	21.3	76.3	63.1	13.2
Phi-4-Mini	65.8	57.1	8.7	79.8	65.0	14.8	67.2	49.1	18.1	60.1	42.4	17.7	68.2	53.4	14.8
Qwen2.5-1.5B	67.1	45.9	21.2	62.6	40.8	21.8	57.0	33.2	23.8	57.0	27.8	29.2	60.9	36.9	24.0
Qwen2.5-3B	74.2	60.8	13.4	77.6	53.8	23.8	71.9	48.2	23.7	58.1	31.0	27.1	70.4	48.5	22.0
Qwen2.5-7B	82.4	78.5	3.9	87.8	72.2	15.6	81.4	67.8	13.6	72.5	51.4	21.1	81.0	67.5	13.5
Average	68.8	57.4	11.4	73.3	54.8	18.5	67.9	48.0	19.9	61.9	36.2	25.7	68.3	49.7	18.6

Table 5: Accuracy (%) of seven SLMs on four commonsense benchmarks, evaluated using o1-mini as an answer extractor (ext.) and as a judge. Δ represents the performance inflation introduced by answer-only evaluation.

(a) Human-annotated step error positions.



(b) o1-mini step error positions.

Figure 3: Comparison of step error positions: (a) human annotation and (b) o1-mini employed as a judge.

standard answer extraction, achieving +5.8 points in accuracy (correctly distinguishing valid from invalid traces) and +20.6 points in error recall (identifying flawed traces) on average. This emphasizes the importance of process-aware evaluation for faithful SLM assessment.

Building on this finding, we evaluate multiple SLMs under both paradigms to quantify the discrepancy between answer-only and reasoning-aware evaluation. Table 5 reveals a consistent 18.6 percentage point average drop when using reasoning-aware evaluation, demonstrating how traditional metrics overestimate SLM capabilities. Even high-performing models like Qwen2.5-7B-Instruct show substantial drops (81.0% to 67.5%).

These results align with our benchmark analysis (Section 3.6) showing that 17.9% of instances reach correct answers through flawed reasoning, and reinforce that o1-mini as a judge better aligns with human assessments. These findings underscore the critical need for reasoning-aware evaluation frameworks that move beyond final answer correctness to accurately reflect SLM reasoning capabilities.

6 Conclusions

In this work, we introduced RETRACEQA, a new gold benchmark for evaluating reasoning traces of SLMs through step-level annotations, including error category locations and categorizations. Our manually annotated benchmark reveals that standard answer-only metrics consistently overestimate SLM performance: on average, 17.9% of the time, SLMs arrive at correct answers via reasoning that contains at least one significant error. Moreover, introducing reasoning-aware evaluation shows that their scores are inflated by up to 25%. Our manual error analysis shows that SLMs struggle primarily with factual grounding (hallucinations account for 41.9-62.5% of errors), though logical coherence issues are also significant (27.9-35.4%).

Although our work demonstrates that LLMs are strong judges and can distinguish correct vs. incorrect traces effectively, they still struggle with error localization. Additionally, PRMs trained on math reasoning fail to transfer to commonsense tasks, highlighting domain-specific gaps and the need for non-math process-level benchmarks like RETRACEQA. We hope that RETRACEQA will encourage broader adoption of reasoning-aware evaluation protocols, thereby providing more reliable assessments of language models.

Limitations

Our work provides valuable insights into reasoning trace evaluation for commonsense reasoning, though some limitations should be taken into account. First, our benchmark focuses exclusively on English-language commonsense reasoning tasks. Extending this evaluation framework to multilingual settings would be valuable for understanding whether reasoning patterns and error distributions vary across languages. Second, while we selected four diverse datasets for commonsense reasoning and extended reasoning trace evaluation beyond mathematics and science, it would be valuable to extend current work on benchmarks capturing reasoning patterns required in other domains, such as procedural reasoning or narrative comprehension. Future work should extend this evaluation framework to a broader range of reasoning modalities in order to establish more comprehensive benchmarks. Finally, our results demonstrate that PRMs trained on mathematical reasoning transfer poorly to commonsense domains, with performance degrading substantially. This domain transfer limitation suggests that reasoning evaluation techniques may require domain-specific adaptations rather than assuming general transferability across reasoning tasks, motivating the need to develop specialized PRMs for other domains beyond mathematics and science.

Overall, while our work provides a solid foundation for reasoning trace evaluation, addressing the aforementioned limitations will be crucial for advancing the field and developing more robust reasoning models.

Acknowledgments

Roberto Navigli and Simone Conia gratefully acknowledge the support of the PNRR MUR project PE0000013-FAIR. Simone's fellowship is fully funded by this project.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Google DeepMind. 2025. Gemini 2.0 flash. https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash. Accessed: 2025-05-10.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, and Qihao Zhu. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. *Preprint*, arXiv:2301.12726.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, and Artem Korenev. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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. 2025. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-star: Training verifiers for self-taught reasoners. *Preprint*, arXiv:2402.06457.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. *Preprint*, arXiv:2310.01798.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. *Preprint*, arXiv:1910.11473.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

J.R. Landis and G.G. Koch. 1977. Measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Jiacheng Liu, Ramakanth Pasunuru, Hannaneh Hajishirzi, Yejin Choi, and Asli Celikyilmaz. 2023. Crystal: Introspective reasoners reinforced with self-feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11557–11572, Singapore. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Mistral. 2025. Mistral small 3. `https://mistral.ai/news/mistral-small-3`. Accessed: 2025-05-10.

Francesco Maria Molfese, Simone Conia, Riccardo Orlando, and Roberto Navigli. 2024. ZEBRA: Zero-shot example-based retrieval augmentation for commonsense question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22429–22444, Miami, Florida, USA. Association for Computational Linguistics.

Francesco Maria Molfese, Luca Moroni, Luca Gioffré, Alessandro Scirè, Simone Conia, and Roberto Navigli. 2025. Right answer, wrong score: Uncovering the inconsistencies of LLM evaluation in multiple-choice question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18477–18494, Vienna, Austria. Association for Computational Linguistics.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, and Akila Welihinda. 2024a. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, and Alex Karpenko and. 2024b. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Sarah Pan, Vladislav Lialin, Sherin Muckatira, and Anna Rumshisky. 2023. Let's reinforce step by step. *Preprint*, arXiv:2311.05821.

Raffaele Pisano and Roberto Navigli. 2026. Process reward models meet planning: Generating precise and scalable datasets for step-level rewards. In *Association for Computational Linguistics: ACL 2026*, San Diego, California. Association for Computational Linguistics.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, and Dayiheng Liu and. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Skywork. 2024. Skywork-o1 open series. `https://huggingface.co/Skywork`.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. 2024. LLMs cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13894–13908, Bangkok, Thailand. Association for Computational Linguistics.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *Preprint*, arXiv:2211.14275.

Fali Wang, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, Tzuhao Mo, Qiuhao Lu, Wanjing Wang, Rui Li, Junjie Xu, Xianfeng Tang, Qi He, Yao Ma, Ming Huang, and Suhang Wang. 2024a. A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness. *Preprint*, arXiv:2411.03350.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2024b. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *Preprint*, arXiv:2312.08935.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Qingchen Yu, Zifan Zheng, Shichao Song, Zhiyu Li, Feiyu Xiong, Bo Tang, and Ding Chen. 2025. xfinder:

Large language models as automated evaluators for reliable evaluation. *Preprint*, arXiv:2405.11874.

Zhongshen Zeng, Pengguang Chen, Shu Liu, Haiyun Jiang, and Jiaya Jia. 2024a. Mr-gsm8k: A meta-reasoning benchmark for large language model evaluation. *Preprint*, arXiv:2312.17080.

Zhongshen Zeng, Yinhong Liu, Yingjia Wan, Jingyao Li, Pengguang Chen, Jianbo Dai, Yuxuan Yao, Rongwu Xu, Zehan Qi, Wanru Zhao, Linling Shen, Jianqiao Lu, Haochen Tan, Yukang Chen, Hao Zhang, Zhan Shi, Bailin Wang, Zhijiang Guo, and Jiaya Jia. 2024b. Mr-ben: A meta-reasoning benchmark for evaluating system-2 thinking in llms. *Preprint*, arXiv:2406.13975.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. The lessons of developing process reward models in mathematical reasoning. *Preprint*, arXiv:2501.07301.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2024. Processbench: Identifying process errors in mathematical reasoning. *Preprint*, arXiv:2412.06559.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

A Solution Generation Prompts

Table 6 and 7 show the prompts used in our work to generate the solutions for the multiple-choice and binary commonsense reasoning benchmarks, respectively. Specifically, we use the standard zero-shot Chain-of-Thought (CoT) prompting strategy (Wei et al., 2023; Kojima et al., 2023) to elicit explicit model reasoning.

B Answer Extractor Details

To ensure that our benchmark contains a balanced number of model outputs reaching a correct or incorrect solution, we use a state-of-the-art LLM-based answer extractor (Yu et al., 2025, xFinder). We select this answer-extraction method rather that relying on simple regular expressions because of its higher agreement with human judgment in scenarios involving free-form text generation (Molfese et al., 2025). Specifically, we adopt xFinder-llama38it,[2] the best-performing variant based on Meta-Llama-3-8B-Instruct. We prompt xFinder by providing it with the input question, the optional set of choices and the model output. We then extract its generated output and compare it against the correct answer. We deem an instance as correct if it reaches the correct answer and incorrect otherwise.

[2] https://huggingface.co/IAAR-Shanghai/xFinder-llama38it

SYSTEM
`You are an expert in commonsense question answering. You are given as input a question and a set of choices. First, provide the reasoning process to answer the question. Finally, provide your final answer.`
USER
`Question: {question}` `Choices: {choices}`

Table 6: Prompt for CSQA, OBQA and QASC datasets.

C Error Examples

Table 8 shows additional examples drawn from our RETRACEQA benchmark. Specifically, the table presents the input problem (consisting of a question and an optional set of choices), the model's reasoning trace divided into paragraphs, and the category of error, annotated according to one of three categories. This table provides a qualitative overview of the errors made by SLMs in the context of commonsense reasoning. In particular, the table highlights problematic reasoning patterns in the models' outputs, with errors annotated following the categorization defined in Section 3.5. The first example represents a misinterpretation error, where the model fails to relate the meaning of the answer choices to the question. The second example shows an hallucination error, where the reasoning traces contains the incorrect statement "bowling alley is not the typical location for throwing a ball at pins". Finally, the third example illustrates a reasoning error in which the model correctly states "A. ecru: This color is very light, almost white color" but then continues with an illogical consequence which is "It would not provide much heat-reflecting capability".

SYSTEM

```
You are an expert in commonsense question answering. You are given as input a yes/no question. First, provide the reasoning process to answer the question. Finally, provide your final answer as 'yes' or 'no'.
```

USER

```
Question: {question}
```

Table 7: Prompt for the StrategyQA dataset.

D Annotation Guidelines

D.1 Task Overview

The goal of our annotation process is to evaluate the step-by-step reasoning traces generated by SLMs in response to questions requiring commonsense reasoning. Annotators are tasked to identify the earliest point in the reasoning trace where an error occurs. Importantly, the final answer produced by the model may be correct even if the intermediate reasoning steps are flawed. As such, annotations focus solely on the reasoning trace rather than final output alone. Each annotation instance includes the question, optional answer choices (for multiple-choice tasks), the ground truth answer, and the model-generated reasoning trace broken into discrete steps.[3] In some cases, additional contextual facts drawn from the original datasets are provided to assist the annotator.

D.2 Annotation Objective

Annotators are instructed to read each step in the reasoning trace and identify the first step that contains an error. The task is framed as a classification problem, where annotators assign an integer index to indicate the position of the first erroneous step. Step indices are zero-based (i.e., 0 refers to the first step), and a value of -1 is used to denote that all steps in the reasoning trace are correct. To minimize the inherent subjectivity of commonsense judgment and provide a reproducible framework for evaluation, annotators apply a hierarchical decision tree. This ensures that every reasoning failure is categorized using one of three mutually exclusive labels based on the structural level of the error: misinterpretation, hallucination and reasoning.

[3]In the following, we use the terms "steps" and "paragraphs" interchangeably.

D.3 Error Definition

We define three categories of errors based on the structural stage at which the reasoning process fails. To remove ambiguity, annotators are instructed to evaluate these categories hierarchically, ensuring that the primary cause of the failure is identified at the most fundamental level (from grounding to inference).

Misinterpretation (Grounding Level). This is the primary level of the taxonomy, representing errors where the model fails to correctly ground its reasoning in the provided input or task constraints. By identifying these errors first, we isolate failures in task comprehension from failures in knowledge or logic. This includes:

- Misinterpreting the core question objective or task requirements.
- Misrepresenting previous reasoning steps or established premises.
- Referencing non-existent answer choices or providing multiple answers when only one is requested.

Example: For the question "Why would you take a bus to work?" with choice A being "commute," a model ruling out this correct option because "the question asks why someone would take a bus, not what a bus is used for" demonstrates misinterpretation of the question's objective.

Hallucination (Content Level). If the input is correctly grounded, we evaluate the accuracy of the external knowledge introduced. This category is strictly reserved for the introduction of empirically false or unverifiable world knowledge. It focuses on the *factual accuracy* of the information used in the trace, independent of the logical structure. This includes:

- Introduction of incorrect facts or assumptions that are not generally valid.
- Generation of hallucinated information that is not inferable from the question or context.

Example: A model stating "rejection is the most likely outcome of an interview" presents an incorrect fact that is not generally valid, constituting a hallucination error.

Reasoning (Inference Level). This category addresses errors in the logical transitions between or within steps. It is applied only when the grounding and factual content are correct, but the model connects them using an invalid logical leap or contradictory inference. This ensures that logical failures are evaluated as pure errors in deduction. This includes:

- Logically unsound or commonsense-violating inferences.
- Contradictory or internally incoherent reasoning.
- Ruling out the correct option despite valid intermediate steps.

Example: For the question "Where spiders might be found among tools?", a model stating "a garage may store tools" but then ruling out garage as "not the most likely place" with unsound reasoning commits a reasoning error.

D.4 Non-errors

Not all irregularities in reasoning traces qualify as errors. Annotators are explicitly instructed *not* to flag the following as erroneous:

- Minor grammatical issues or unusual phrasing that do not affect semantic content.
- Verbose, redundant, or overly detailed reasoning that remains logically sound.

D.5 Annotation Procedure

The annotation process involves six key steps:

1. Read the question, any associated answer choices, and any additional supporting facts.
2. Consult the ground truth answer to understand the correct resolution.
3. Examine each reasoning step in sequence.
4. Determine whether each step is sound following the provided guidelines.
5. Record the index of the first erroneous step, or -1 if all steps are correct.
6. Categorize the error (hallucination, reasoning, or misinterpretation) if an error is found.

Importantly, annotators are instructed to mark only the earliest point in the trace where an error occurs, as later steps may be incorrect solely due to propagation from a previously erroneous step (Lightman et al., 2023; Zheng et al., 2024). This process ensures that annotations are consistent, fine-grained, and focused on evaluating the internal validity of reasoning traces rather than their final outcomes alone.

D.6 Identifying Invalid Instances

During the annotation process, some instances may be identified as problematic due to flaws in the original question or labeling. To maintain the quality of the benchmark and avoid propagating errors from upstream datasets, annotators are asked to flag such examples with a special INVALID tag. The following conditions qualify an instance for exclusion: (i) the question contains severe grammatical issues that compromise its interpretation; (ii) multiple answer choices are semantically identical or equally valid; (iii) the dataset's annotated ground truth is demonstrably incorrect based on commonsense or factual knowledge; and (iv) the instance lacks a unique, clearly correct answer. Instances tagged as INVALID are excluded from the analysis.

E Individual Model Statistics

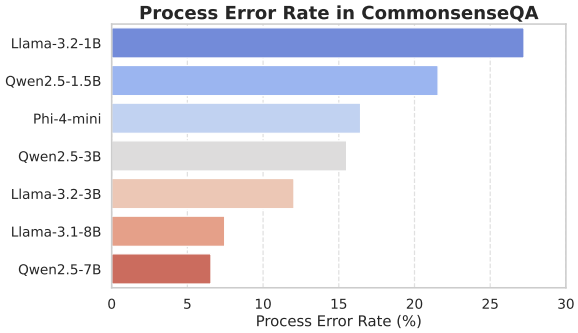
Figure 4 shows the individual process error rates of the seven SLMs listed in Section 3.3 (we omit "Instruct" tags for readability), measured as the percentage of instances in which an incorrect reasoning trace leads to the correct final answer. In particular, we can see that the trend is approximately the same for each subset of RETRACEQA: smaller models correspond to higher process error rates, with the latter decreasing as model size increases.

F LLM-as-a-judge Prompts

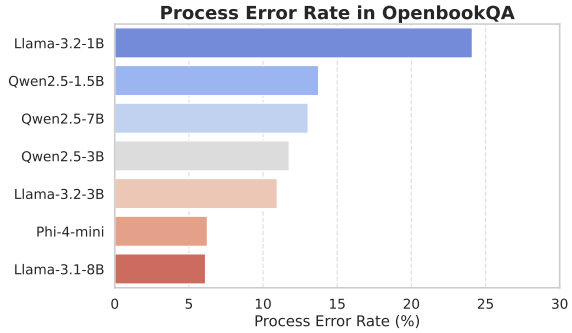
Table 9 show the prompts used for LLM-as-a-judge evaluation of commonsense reasoning traces. They are lightly adapted from prior work (Zheng et al., 2024), omitting answer choices when unavailable and including supporting facts when provided.

Input	Reasoning Trace	Error Category
Question: birds use their peckers to catch **Choices**: A. dogs B. a tan C. a ball D. bees	**Paragraph 0:** The question asks what birds use their peckers for. [...] **Paragraph 5:** The choices provided are not related to the question at hand.	Misinterpretation.
Question: Where do you throw a ball at pins? **Choices**: A. bowling alley B. football stadium C. soccer field D. sporting event E. sporting goods store	**Paragraph 0:** To answer this question, we need to consider the context and the typical location of a ball-throwing activity. [...] **Paragraph 2:** A bowling alley is primarily used for bowling, a specific type of ball-throwing game. It is not the typical location for throwing a ball at pins.	Hallucination.
Question: What is a more comfortable color to have for your automobile upholstery if living in a desert? **Choices**: A. ecru B. red C. black D. navy	**Paragraph 0:** To answer this question, we need to consider the climate and environment of a desert. Deserts typically have high temperatures and intense sunlight, so the upholstery color should ideally have good heat-reflecting properties to keep the car interior cooler. [...] **Paragraph 2:** A. ecru: This color is a very light, almost white color. It would not provide much heat-reflecting capability.	Reasoning.

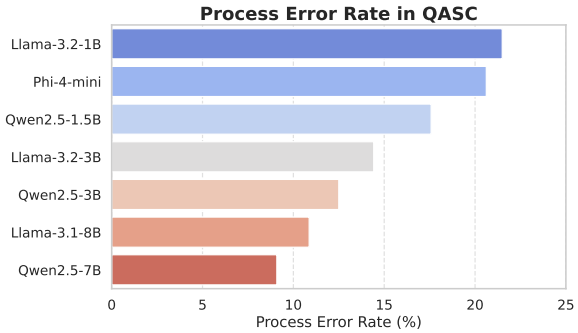
Table 8: Additional examples drawn from our RETRACEQA benchmark.



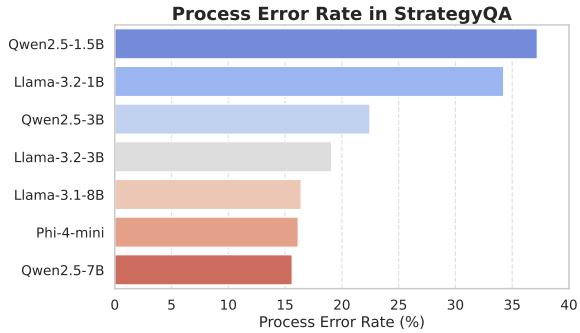
(a) CSQA Process Error Rate



(b) OBQA Process Error Rate



(c) QASC Process Error Rate



(d) StrategyQA Process Error Rate

Figure 4: Process Error Rate (%): The proportion of incorrect reasoning traces that reach the correct final answer, calculated across the annotated subsets of our benchmark for each model.

SYSTEM

You are an expert in carefully analyzing step-by-step solutions for commonsense reasoning problems.

USER

The following is a commonsense reasoning problem composed of a question, a set of choices, the correct answer and a solution (split into paragraphs, enclosed with tags and indexed from 0):

[Commonsense Problem]

Question: {question}

Choices: {choices}

Answer: {answer}

[Solution]

{model_output}

Your task is to review and critique the solution paragraph by paragraph. Once you identify a commonsense reasoning error in a paragraph, return the index of the paragraph where the earliest error occurs. Otherwise, return the index of -1 (which typically denotes 'not found').

Please put your final answer (i.e., the index) in boxed{}.

Table 9: LLM-as-a-judge prompt.