Chain-of-Thought as a Lens: Evaluating Structured Reasoning Alignment between Human Preferences and Large Language Models

Boxuan Wang, Zhuoyun Li, Xinmiao Huang, Xiaowei Huang, Yi Dong*
School of Computer Science and Informatics, University of Liverpool, United Kingdom
{boxuan.wang, zhuoyun.li, xinmiao.huang, xiaowei.huang, yi.dong}@liverpool.ac.uk

Abstract

This paper primarily demonstrates a method to quantitatively assess the alignment between *multi-step, structured reasoning* in large language models and human preferences. We introduce the Alignment Score, a semantic-level metric that compares a model-produced chain of thought traces with a human-preferred reference by constructing semantic-entropy-based matrices over intermediate steps and measuring their divergence. Our analysis shows that Alignment Score tracks task accuracy across models and hop depths, and peaks at 2-hop reasoning. Empirical results further indicate that misalignment at greater reasoning depths is driven mainly by alignment errors such as *thematic shift* and *redundant reasoning*. Viewing chain sampling as drawing from a distribution over reasoning paths, we empirically demonstrate a strong and consistent correlation between Alignment Score and accuracy, readability, and coherence, supporting its use as a diagnostic signal. The code is available.[1]

1 Introduction

In-Context Learning (ICL) is widely regarded as a core capability of Large Language Models (LLMs), allowing them to tackle diverse tasks by conditioning on a few example instances included in the input prompt, all without modifying their parameters during inference (Dong et al., 2024). A particular approach within ICL is Chain-of-Thought (CoT) prompting (Wei et al., 2022), which encourages models to generate intermediate reasoning steps to achieve a final answer. CoT has been shown to significantly enhance model performance in complex reasoning tasks, e.g., arithmetic reasoning (Cobbe et al., 2021), symbolic logic (Zhou et al., 2023), and commonsense question answering (Kojima et al., 2022). Although CoT and its established variants

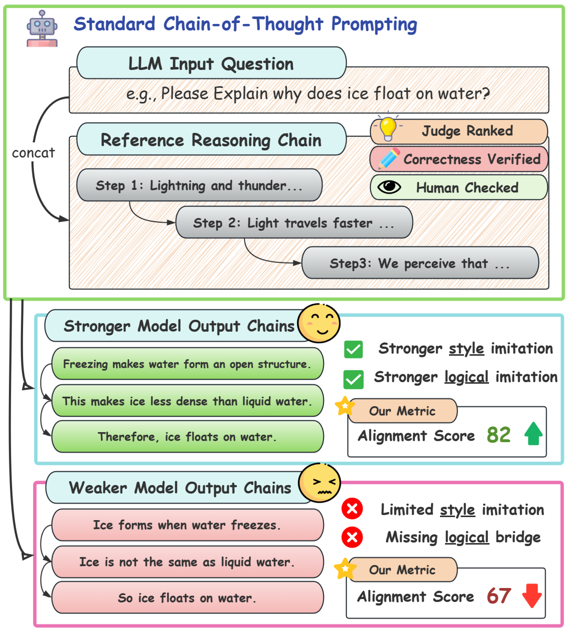

Figure 1: An illustration of comparing reasoning consistency in strong vs. weak models under CoT prompting: the proposed Alignment Score quantifies their divergence, with higher values indicating the stronger style imitation and improved logical coherence.

(e.g., Least-to-most (Zhou et al., 2023) and self-consistency (Wang et al., 2023)) have led to gains in both robustness and accuracy, recent work shows that LLM reasoning traces still vary substantially in quality (Turpin et al., 2023; Hu et al., 2025). In particular, structured or multi-step reasoning often suffers from semantic incoherence, logical inconsistency, or thematically misaligned steps, even when the final answer is correct (Lampinen et al., 2022; Kojima et al., 2022). These findings expose a key gap: we still lack an evaluation metric that goes beyond answer correctness and captures the quality of the reasoning process itself.

As illustrated in Figure 1, to address this gap, we revisit multi-step reasoning in CoT through the

*Corresponding Author: yi.dong@liverpool.ac.uk
[1] https://github.com/boxuanwang28/CoT-Lens

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 39514–39530
July 2-7, 2026 ©2026 Association for Computational Linguistics

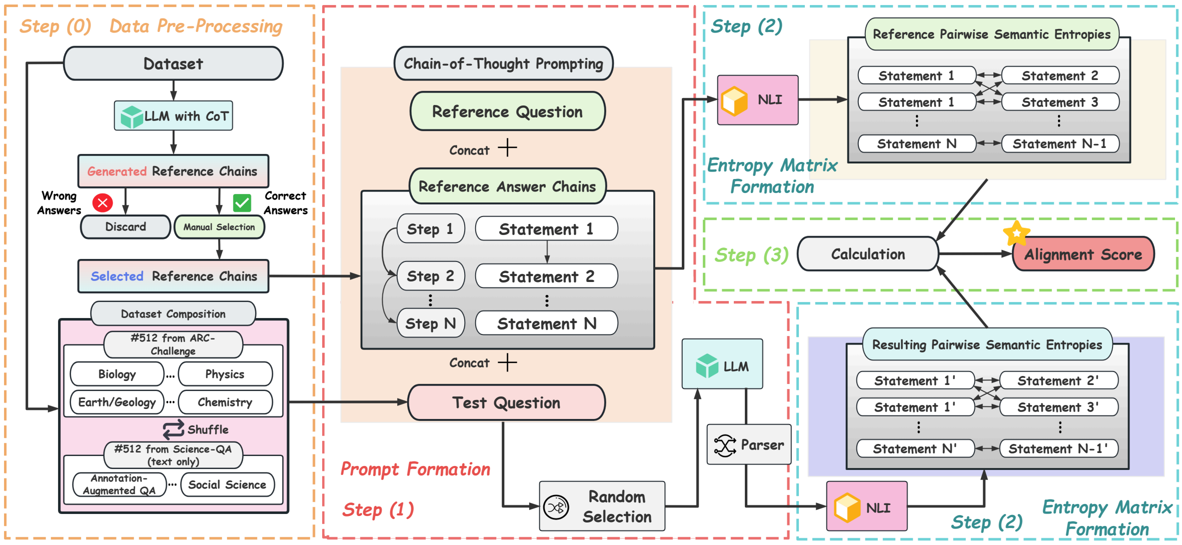

Figure 2: Illustration of the Alignment Score Computation: **Step (0)**: Prepare the dataset and select reference chains (orange box). **Step (1)**: Construct the prompt with a reference reasoning chain and generate model reasoning chains (red box; see Appendix A for prompt templates); **Step (2)**: Compute pairwise semantic entropy matrix using an NLI model (blue box); **Step (3)**: Compare the resulting entropy matrix and derive the Alignment Score (green box).

perspective of *reasoning consistency*.[2] Our major hypothesis is that failures in reasoning performance often stem not from the inability of models to compute correct answers, but from the misalignment between their generated reasoning chains and reference patterns that exhibit coherent, human-like logical structures.

Building on this motivation, we design an Alignment Score framework that treats CoT as the main handle for measuring how well model reasoning aligns with human-preferred chains, as shown in Figure 2. We first manually filter model-generated CoT explanations to obtain a pool of correct, well-structured reference chains. These references are inserted as in-context exemplars for each test question, prompting the model to imitate their reasoning style. We then compare the reference chain and the generated chain in a latent space defined by semantic entropy (Kuhn et al., 2023): pairwise judgments form entropy matrices over reasoning steps, and their divergence is summarized as a single Alignment Score that reflects structural and logical, rather than surface-level, agreement.

To interpret this score and link it to task performance, we analyze how specific alignment errors and reasoning depth affect both Alignment Score and accuracy. From empirical inspection, we define two interpretable error types, *Thematic Shift* and *Redundant Reasoning*, and track how their frequencies evolve with the number of reasoning hops. Finally, we introduce two alignment-aware sampling schemes, Alignment-aware Chain Sampling and Selection (ACSS) and Self-Consistency with Alignment (SC-Align), which sample multiple CoT chains under a fixed budget and then select chains using alignment-based criteria, allowing us to probe whether higher Alignment Scores correspond to higher accuracy and better reasoning quality. Our main contributions are threefold:

(1) We propose the **Alignment Score**, a metric that quantitatively measures how closely model-generated reasoning traces align with human-preferred reference chains, and validate its effectiveness and reliability through comparison-based correlation analysis.

(2) We conduct a detailed empirical study of how reasoning depth influences Alignment Score, and show that *Thematic Shift* and *Redundant Reasoning* emerge as dominant alignment errors as reasoning depth increases.

(3) We present **ACSS** and **SC-Align** as alignment-aware sampling methods, with two main findings: (i) Alignment Score closely tracks task accuracy; and (ii) Alignment Score is strongly associated with improved readability and coherence under extensive LLM-based evaluations.

[2] *Reasoning consistency* refers to the extent of logical congruence and step-by-step agreement between the model's generated reasoning chains and human-preferred reference chains (i.e., coherent and readable traces with correct final answers).

2 Related Work

LLM Evaluation Techniques. Evaluating the quality of LLM outputs has long been a central topic. Existing answer-focused benchmarks such as MMLU (Hendrycks et al., 2021a), ARC-AGI (Chollet, 2019), GPQA (Rein et al., 2024), GSM8K (Cobbe et al., 2021), TruthfulQA (Lin et al., 2022), and the broad BIG-Bench suite (Srivastava et al., 2023) together with its harder subset BBH (Suzgun et al., 2023), cover a wide range of tasks but largely ignore whether *reasoning process* is plausible or self-consistent. To bridge this gap, several evaluation frameworks specifically targeting CoT have recently emerged. REVEAL (Jacovi et al., 2024) provides fine-grained step-level annotations to measure logical correctness. The CoTKG paradigm of Nguyen *et al.* (Nguyen et al., 2024) directly grounds intermediate steps in knowledge graphs (KGs), producing discriminative and generative scores for multi-hop reasoning. Vacareanu *et al.* (Vacareanu et al., 2024) further introduces a general-purpose verifier that checks each step for relevance, mathematical accuracy, and logical consistency. Our approach differs from the above by evaluating LLMs from the perspective of reasoning consistency between generated reasoning chains and reference CoT demonstrations. Unlike REVEAL and CoTKG, our evaluation method does not rely on step-level annotations, external verifiers, or knowledge graph construction, making it more lightweight and broadly applicable.

Semantic Entropy. Semantic-level uncertainty methods aim to capture variability over *meanings* rather than surface forms. Semantic Entropy has emerged as an effective approach for estimating LLM uncertainty by sampling outputs, clustering semantically equivalent responses via entailment, and computing entropy over the induced meaning distribution (Farquhar et al., 2024). Building on this idea, Kernel Language Entropy (KLE) uses kernels over semantic similarities to obtain smoother entropy estimates (Nikitin et al., 2024), Shapley Uncertainty decomposes semantic variability via Shapley-value attributions (Zhu et al., 2025), and Semantic Energy combines Boltzmann-style energy with semantic clustering for confidence estimation (Ma et al., 2025). In this work, we apply semantic entropy in a *one-to-one* alignment setting, directly measuring semantic dispersion between each generated reasoning step and its corresponding reference step. To our knowledge, this constitutes the first systematic use of semantic entropy for evaluating *step-structured reasoning* in LLMs.

3 Methodology

In this section, we first present the necessary preliminaries. Then, we present the calculation of the Alignment Score. Finally, we introduce the methods of alignment-aware chain selection.

3.1 Preliminaries

Pairwise Semantic Entropy Semantic entropy has been proposed as a measure of semantic uncertainty, practically based on Natural Language Inference (NLI) entailment signals, (Kuhn et al., 2023; Farquhar et al., 2024; Vashurin et al., 2025; Kossen et al., 2024). Given a pair of statements (S_i, S_j), the semantic relationship: entailment (E), contradiction (C), or neutrality (N), is predicted by an NLI model, which gives a probability distribution $\{p_E, p_C, p_N\}$. The pairwise entropy $H(S_i, S_j)$ can then be computed as:

$$H(S_i, S_j) = -\sum_{k \in \{E,C,N\}} p_k \log_2(p_k) \quad (1)$$

While prior CoT evaluation frameworks (Jacovi et al., 2024; Nguyen et al., 2024) typically assess local sentence pairs in isolation (e.g., an intermediate claim and its premise), our goal is different: we seek to evaluate the internal semantic coherence within a reasoning chain, as well as the semantic consistency between two chains. To this end, we propose a pairwise formulation of semantic entropy over all step pairs in a reasoning path.

In this scenario, a reasoning chain can be presented as $\mathcal{C} = \{S_1, S_2, \ldots, S_n\}$, where S_i is the i-th statement in the reasoning chain. Similarly, the reference reasoning chain can be denoted as $\mathcal{C}^{\text{ref}} = \{S_1^{\text{ref}}, S_2^{\text{ref}}, \ldots, S_n^{\text{ref}}\}$, where the superscript "ref" indicates the human-preferred reference chain. For each pair of statements (S_i, S_j) where $i, j \in \{1, 2, \ldots, n\}$ and $i \neq j$, we compute the pairwise semantic entropy as:

$$H_{i,j} = H(S_i, S_j) \quad (2)$$

For convenience, we represent the semantic entropy between two statements by $H(i, j)$. This pairwise form enables us to quantify semantic consistency both within and between reasoning chains.

Although LLM outputs are not inherently structured as strict stepwise sequences, recent studies have shown that appropriate prompting strategies and ICL techniques (Dong et al., 2024) can effectively elicit multi-step reasoning behaviors from LLMs. In our setting, we assume that the reasoning output has been formatted into a list of steps through CoT prompting with chain-annotated demonstrations. Given n reasoning steps, there are $\binom{n}{2} = \frac{n(n-1)}{2}$ unique pairs to evaluate, forming the basis for constructing the semantic entropy.

Semantic Entropy Matrix The pairwise semantic entropy values can be organized in matrix form for further structural comparison. There will be two matrices: one for model-generated reasoning chains, and one for human-written reference chains, as illustrated in the two blue boxes of Figure 2.

Given a model output reasoning chain $\mathcal{C} = \{S_1, S_2, \ldots, S_n\}$, we construct its semantic entropy matrix $\mathbf{H}$ as:

$$\mathbf{H} = \begin{bmatrix} 0 & H_{1,2} & H_{1,3} & \cdots & H_{1,n} \\ H_{2,1} & 0 & H_{2,3} & \cdots & H_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ H_{n,1} & H_{n,2} & \cdots & 0 & \end{bmatrix} \tag{3}$$

Similarly, for the reference reasoning chain $\mathcal{C}^{\text{ref}} = \{S_1^{\text{ref}}, S_2^{\text{ref}}, \ldots, S_n^{\text{ref}}\}$, we can construct its corresponding semantic entropy matrix $\mathbf{H}^{\text{ref}}$.

3.2 Calculation of Alignment Score

To measure how far the model's reasoning structure deviates from the reference, we first compute a *semantic divergence* D_{sem} between their normalized entropy matrices. Concretely, we take the upper-triangular entries of the reference entropy matrix and the model entropy matrix $\mathbf{H}^{\text{ref}}$ and $\mathbf{H}$, normalize them into two probability distributions $\mathbf{p}^{\text{ref}}$ and $\mathbf{p}$. Here, $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler (KL) divergence between two probability distributions. We then define:

$$D_{\text{sem}} = \tfrac{1}{2}\,\mathrm{KL}\Big(\mathbf{p}^{\text{ref}} \,\Big\|\, \tfrac{\mathbf{p}^{\text{ref}}+\mathbf{p}}{2}\Big) + \tfrac{1}{2}\,\mathrm{KL}\Big(\mathbf{p} \,\Big\|\, \tfrac{\mathbf{p}^{\text{ref}}+\mathbf{p}}{2}\Big) \tag{4}$$

We adopt the Jensen–Shannon (JS) divergence as our semantic divergence because it is bounded, and more numerically stable than plain KL divergence when the model assigns low probability mass to regions where the reference is confident (Lin, 1991; Li et al., 2016). A smaller D_{sem} indicates that the model-induced semantic distribution P is closer to the reference distribution P^{ref} (i.e., smaller information mismatch) and thus reflects stronger stepwise semantic agreement and less misalignment. We then monotonically map D_{sem} to a score in $[0, 100]$, as detailed in Algorithm 1.

Algorithm 1 Alignment Score Calculation

Require: Dataset of questions with reference chains $\{\mathcal{R}_q\}$, model-generated chains $\{\mathcal{C}_q\}$, entropy encoder
Ensure: Average Alignment Score $\bar{S} \in [0, 100]$
1: Initialize score list $\mathcal{S} \leftarrow [\,]$
2: **for** each question q **do**
3: Build reference entropy matrix H^{ref} from $\mathcal{R}_q$ and obtain normalized upper-triangular vector $\mathbf{p}^{\text{ref}}$
4: **for** each model chain $c \in \mathcal{C}_q$ **do**
5: Build entropy matrix H from c and obtain normalized upper-triangular vector $\mathbf{p}$
6: Let $\mathbf{m} \leftarrow (\mathbf{p}^{\text{ref}} + \mathbf{p})/2$ and compute $D_{\text{sem}} \leftarrow \frac{1}{2}\mathrm{KL}(\mathbf{p}^{\text{ref}}\|\mathbf{m}) + \frac{1}{2}\mathrm{KL}(\mathbf{p}\|\mathbf{m})$
7: $S \leftarrow (1 - D_{\text{sem}}) \times 100$; append S to $\mathcal{S}$
8: **end for**
9: **end for**
10: **return** $\bar{S} = \dfrac{1}{|\mathcal{S}|}\sum_{S\in\mathcal{S}} S$

3.3 Alignment Errors

To better understand how our Alignment Score relates to concrete failure modes and downstream task performance, we explicitly define a small set of *alignment errors* that are both quantifiable and easy to interpret. From empirical observations, two types of errors appear most frequently: *Thematic Shift* (TS) and *Redundant Reasoning* (RR). The two error types are defined as follows.

Thematic shift (TS). TS captures changes in topic or theme across adjacent reasoning steps. Significant thematic drift suggests that the model has lost coherence or deviated from the core reasoning path (Golovneva et al., 2023). We detect shifts by computing the cosine similarity between sentence embeddings of consecutive steps and counting how many times it falls below a threshold τ_{TS}. Let e_{B_i} denote the embedding of the i-th step:

$$\text{TS} = \sum_{i=1}^{n-1} \mathbb{I}\big(\cos(e_{B_i}, e_{B_{i+1}}) < \tau_{\text{TS}}\big). \tag{5}$$

Redundant reasoning (RR). RR measures *semantic* repetition in reasoning chains, rather than exact string duplication. High redundancy indicates that the model keeps "circling around" the same point instead of making genuine progress (Chiang and Lee, 2024; Jang et al., 2025). Let e_{B_i} be the embedding of step B_i. For each step, we check whether it is overly similar to *any* previous step:

$$\text{RR} = \sum_{i=2}^{n} \mathbb{I}\Big(\max_{1\le j<i} \cos(e_{B_i}, e_{B_j}) > \tau_{\text{RR}}\Big), \tag{6}$$

where τ_{RR} is a high similarity threshold. A step is thus marked as redundant if its semantic content is almost entirely covered by an earlier step, even when the surface wording differs.

Thresholds τ_{TS} and τ_{RR} are selected via percentile-based calibration based on a development set. A summary can be found in Appendix B.

3.4 Alignment-Aware Chain Selection

We use test-time compute to probe how task accuracy, Alignment Score, and alignment errors are related. For each question, we sample K reasoning chains under the same decoding budget, and only vary how we select a single final chain. We propose the following two chain selection strategies:

Alignment-Aware Chain Sampling and Selection (ACSS). In the first strategy, we select one chain from the K sampled chains following the *Best-of-N* (Stiennon et al., 2020) paradigm. Concretely, we either minimize the number of alignment errors (e.g., thematic shift or redundant reasoning) or directly maximize the Alignment Score over the K chains. This simple scheme makes it easy to see how strongly these alignment signals correlate with downstream task accuracy.

Self-Consistency with Alignment (SC-Align). In the second strategy, we further ask whether chains that are more likely to give the correct answer naturally have higher Alignment Scores. To test this, SC-Align follows the *Self-Consistency* (Wang et al., 2023) paradigm: we first apply majority voting over the K chains to obtain the subset of chains that support this answer, then select the chain with the highest Alignment Score as the final explanation. This keeps the same test-time compute as self-consistency, while directly testing whether Alignment Score can serve as a practical handle for improving performance.

3.5 LLM-based Preference Evaluation

To check whether higher Alignment Scores match human preferences, we use an *LLM-as-a-judge* paradigm (Huang et al., 2025; Bai et al., 2024) to avoid human subjectivity and potential bias. For each question, the judge model is given the problem, answer options, the human reference chain, and two candidates: the baseline chains and the SC-Align chains. It assigns 1–10 scores for *coherence* (logical completeness and consistency) and *readability*, and outputs a pairwise preference for each aspect (which chain is better or tie) with brief explanations. Detailed configurations of the evaluation pipeline can be found in Appendix C.

4 Experiments

This section presents a series of experiments to answer the following research questions (RQs):

RQ1: Is the Alignment Score a reliable metric for evaluating reasoning alignment?

RQ2: How does the depth of reasoning (i.e., reasoning hop) affect the Alignment Score?

RQ3: How strongly is the Alignment Score correlated with task performance (accuracy)?

In the following sections, we first describe the experimental settings and how the reference chains are constructed, and then address each RQ in turn.

4.1 Experimental Setup

Dataset. We carry out our experiments on a fused evaluation set of 1,024 multiple-choice questions drawn from two benchmarks: the text-only subset of ScienceQA (Lu et al., 2022) and the ARC-Challenge benchmark (Clark et al., 2018).

Models. Our experiments consider LLMs spanning open-source, closed-source, and reasoning-oriented families, covering parameter scales from a few billion to the hundred-billion level. The open-source models are Falcon-7B-Instruct (Technology Innovation Institute, 2023), Qwen2.5-3B-Instruct (Yang et al., 2025), and LLaMA2-13B-Chat (Touvron et al., 2023). The closed-source non-reasoning models are GPT-3.5-turbo (OpenAI, 2023), GPT-4o-mini (OpenAI, 2024b), and GPT-4o (OpenAI, 2024a), accessed via the OpenAI API. The reasoning models are GPT-o1 (OpenAI, 2024c) and DeepSeek-R1 (Guo et al., 2025). Hyperparameter settings can be found in Appendix A.2.

Statistical Reporting. All experiments report the mean and standard deviation over five runs.

Reference Chains Construction. To evaluate semantic alignment, we construct 1,024 reference reasoning chains of varying depths (from 1-hop to 4-hop, 256 for each hop), where intermediate steps lead to the conclusion. We limit our evaluation from 1-hop to 4-hop reasoning for both practical and cognitive reasons. Empirical studies such as HotpotQA (Yang et al., 2018) show that most multi-hop questions need only 1–2 supporting facts, and reasoning beyond 4 hops is rare in real few-shot settings. This practical limitation is further reinforced by cognitive constraints: human working memory

Model Name	Hop-wise Alignment Score & Avg. (Mean ± Std)					Comparisons		
	1-Hop	2-Hop	3-Hop	4-Hop	Avg.	MATH	GPQA	MMLU
Open-Source Models (non-reasoning)								
Falcon-7B-Instruct	75.23 (±26.94)	**79.44 (±16.01)**	73.38 (±13.74)	70.44 (±10.09)	76.02 (±16.70)	0.01	0.25	0.28
Qwen2.5-3B-Instruct	75.93 (±28.90)	**81.64 (±16.06)**	79.51 (±12.86)	77.60 (±11.77)	79.03 (±17.40)	0.76	0.30	0.65
LLaMA2-13B-Chat	77.46 (±27.56)	**79.67 (±17.06)**	77.19 (±13.49)	72.28 (±10.60)	78.10 (±17.18)	0.28	0.23	0.51
Closed-Source Models (non-reasoning)								
GPT-3.5-Turbo (2023-11)	76.97 (±28.09)	**81.00 (±16.59)**	78.50 (±12.91)	77.07 (±10.39)	78.82 (±17.00)	0.43	0.31	0.70
GPT-4o (2025-03)	78.78 (±26.43)	**81.59 (±16.23)**	78.81 (±12.94)	74.59 (±10.95)	79.73 (±16.64)	0.77	0.54	0.86
GPT-4o-Mini (2024-11)	**81.58 (±24.99)**	80.66 (±17.18)	77.99 (±13.04)	74.80 (±10.93)	80.08 (±16.54)	0.70	0.40	0.82
Reasoning Models								
GPT-o1 (2024-09)	80.20 (±25.91)	**82.85 (±15.81)**	79.95 (±11.60)	78.60 (±10.22)	81.00 (±15.89)	0.96	0.78	0.92
DeepSeek–R1 (2025-05)	77.30 (±27.82)	**87.82 (±10.17)**	86.91 (±12.70)	82.87 (±11.06)	84.01 (±15.44)	0.97	0.72	0.91
Pearson Correlation (w.r.t. Avg. Score)						0.88	0.83	0.87

Table 1: Hop-wise Alignment Score and comparison with other benchmarks. **Bold** values indicate the highest Alignment Score for each model. The benchmark scores in the "Comparisons" columns are normalized to the $[0, 1]$ range for the ease of comparison. The date in parentheses indicates the assessed version of closed-source models.

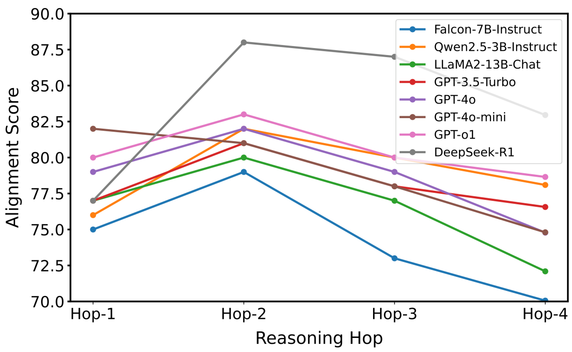

Figure 3: The Alignment Score trend across reasoning hops. Most models peak at 2-hop reasoning.

typically holds only 3–4 "chunks" (Cowan, 2001), making longer reasoning chains more difficult to interpret and construct as reliable reference chains. Detailed process for reference chain construction can be found in Appendix D.

4.2 RQ1: Effectiveness of Alignment Score

To assess the effectiveness of the proposed Alignment Score, we conduct a comparative evaluation across open-source, closed-source, and reasoning-oriented models. Table 1 reports hop-wise Alignment Scores (from 1-hop to 4-hop) and their average, together with three normalized comparing benchmark scores, including MATH (Hendrycks et al., 2021b), GPQA (Rein et al., 2024), and MMLU (Hendrycks et al., 2021a).

Overall Trend. A clear cross-model trend emerges, as illustrated in Figure 3. Within the open-source group, Qwen2.5-3B-Instruct attains the highest average Alignment Score, followed closely by LLaMA2-13B-Chat and then Falcon-7B-Instruct. In the closed-source non-reasoning group, GPT-4o-mini and GPT-4o both substantially outperform GPT-3.5-turbo. Reasoning-oriented models (GPT-o1 and DeepSeek-R1) achieve the highest Alignment Scores overall. This ordering closely mirrors the commonly observed capability hierarchy of these models, indicating that the Alignment Score is sensitive to genuine reasoning strength.

Correlation with Comparison Benchmarks. To further validate this, we compute Pearson correlations between the *average* Alignment Score and comparison benchmarks. The Alignment Score correlates strongly with MATH (**0.88**), GPQA (**0.83**), and MMLU (**0.87**), which respectively target mathematical problem solving, graduate-level scientific reasoning, and broad multi-domain knowledge. Although Alignment Score is computed purely from reference-chain alignment rather than answer accuracy, it tracks benchmark performance closely and thus indicates that it can provide a reliable, task-agnostic signal of overall reasoning quality.

4.3 RQ2: Alignment Score Peaks at 2-Hop

As illustrated in Figure 3, we observe that the Alignment Score peaks at 2-hop reasoning in almost all

Method	Open-Source Models (non-reasoning)			Closed-Source Models (non-reasoning)			Reasoning Models	
	Falcon-7B	Qwen2.5-3B	LLaMA2-13B	GPT-3.5-turbo	GPT-4o-mini	GPT-4o	GPT-o1	DeepSeek-R1
Accuracy (mean ± std, %)								
CoT	23.02 (±1.39)	75.59 (±2.30)	59.75 (±3.90)	74.58 (±2.53)	86.93 (±1.07)	92.68 (±0.60)	**96.12** (±0.12)	**98.88** (±0.07)
SC-CoT	23.07 (±0.99)	**77.71** (±1.80)	61.39 (±2.00)	76.91 (±1.41)	88.16 (±2.89)	93.02 (±0.43)	93.61 (±1.10)	96.36 (±0.12)
ACSS-TS	23.17 (±2.19)	75.59 (±2.30)	60.24 (±1.81)	**77.08** (±2.88)	88.21 (±2.62)	92.78 (±0.59)	93.56 (±2.82)	96.30 (±0.20)
ACSS-RR	22.50 (±2.60)	73.78 (±2.64)	60.11 (±1.60)	75.51 (±1.98)	87.10 (±2.42)	92.94 (±0.78)	94.73 (±1.28)	96.49 (±0.18)
ACSS-Ali.	**25.21** (±3.05)	76.00 (±0.11)	**61.78** (±4.18)	75.81 (±4.38)	**89.92** (±1.03)	**94.59** (±1.71)	**94.92** (±1.40)	**96.53** (±0.40)
SC-Align	**24.02** (±2.83)	**77.24** (±1.99)	**61.58** (±1.75)	**77.40** (±3.31)	**88.81** (±2.75)	**93.02** (±0.64)	94.82 (±1.67)	96.36 (±0.21)
Alignment Score (mean ± std)								
CoT	73.38 (±13.75)	79.18 (±12.46)	77.19 (±13.49)	78.50 (±12.91)	78.18 (±13.58)	78.81 (±12.94)	79.95 (±11.60)	86.91 (±12.70)
SC-CoT	75.84 (±13.44)	79.47 (±12.36)	76.70 (±12.98)	77.61 (±13.64)	78.60 (±13.16)	78.82 (±13.13)	79.02 (±11.76)	86.15 (±10.46)
ACSS-TS	74.89 (±13.30)	79.18 (±12.46)	76.73 (±13.57)	77.72 (±13.29)	77.96 (±13.48)	78.94 (±13.28)	76.45 (±14.99)	87.62 (±14.58)
ACSS-RR	77.52 (±14.09)	79.48 (±12.39)	76.93 (±13.01)	78.22 (±13.04)	78.34 (±13.08)	79.09 (±13.09)	80.21 (±10.12)	87.74 (±14.35)
ACSS-Ali.	**80.15** (±13.08)	**88.28** (±7.85)	**86.71** (±8.82)	**88.05** (±7.91)	**87.45** (±8.50)	**87.90** (±8.37)	**88.15** (±7.46)	**89.42** (±12.37)
SC-Align	**80.97** (±12.61)	**88.33** (±7.65)	**86.00** (±8.92)	**87.34** (±8.78)	**86.79** (±9.22)	**88.07** (±8.36)	**86.40** (±9.90)	**89.58** (±10.33)

Table 2: Cross-model comparison of methods under 3-hop reasoning. Rows with gray, blue, and beige backgrounds correspond to the baselines, ACSS methods and SC-Align, respectively. ACSS-TS, ACSS-RR, and ACSS-Ali. denote chain selection methods that, respectively, minimize thematic shift/redundant reasoning, or maximize the Alignment Score. **Red** numbers indicate the best result for each model, and **blue** numbers the second best.

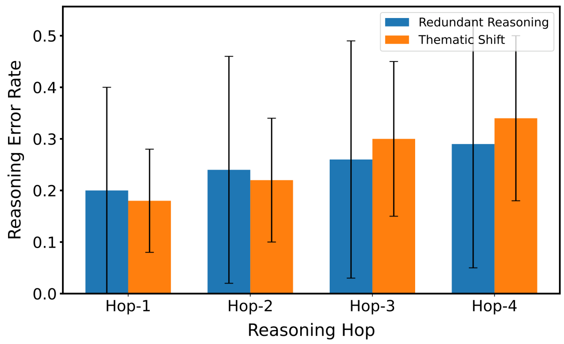

Figure 4: Alignment errors across reasoning hops.

models. In the following, we provide a theoretical and empirical explanation based on the structure of the semantic entropy matrix used in our metric.

1-hop is structurally under-informative. Recall that a reasoning chain $\mathcal{C} = \{S_1, \ldots, S_n\}$ induces a semantic entropy matrix $\mathbf{H}$ (and $\mathbf{H}^{\text{ref}}$ for the reference chain), whose upper-triangular entries are normalized into probability distributions $\mathbf{p}$ and $\mathbf{p}^{\text{ref}}$; the semantic divergence D_{sem} and the Alignment Score are then computed from these distributions (Section 3.2). When reasoning hop $h = 1$, both $\mathbf{p}$ and $\mathbf{p}^{\text{ref}}$ collapse to one-dimensional distribution supported by a single entropy value. In this degenerate scenario, D_{sem} reduces to a single pairwise relation and contains no information about inter-step structure. As a result, the Alignment Score for 1-hop chains is intrinsically less sensitive to reasoning quality, because there is insufficient structural evidence to expose semantic misalignment. This justifies treating Alignment Score at 1-hop as a trivial lower-bound regime rather than a reliable indicator of multi-step reasoning ability, and explains why the Alignment Scores at 1-hop are relatively lower in most cases.

Deeper hops reduce alignment. For $h \geq 2$, the semantic entropy matrix becomes expressive, enabling more meaningful comparison. However, as hop length increases, the number of pairwise relations grows quadratically ($O(h^2)$), and reasoning errors propagate to multiple matrix entries. Consequently, accumulated errors such as thematic drift and redundancy are amplified, leading to a systematic decrease in the score. This effect is empirically reflected in the increased proportion of alignment errors at greater hops, as illustrated in Figure 4.

Trade-off and 2-hop optimality. Taken together, 2-hop reasoning represents an effective trade-off: it is the minimal depth that yields a non-degenerate semantic structure while remaining short enough to limit error accumulation. This balance explains the observed peak of the Alignment Score at 2-hop.

4.4 RQ3: Does Alignment Score Predict and Improve Task Performance?

Table 2 summarizes the cross-model results in terms of task Accuracy and Alignment Score, com-

Method	ΔAccuracy (pp)	ΔAlignment Score
ACSS-TS	−0.41	−0.34
ACSS-RR	−0.88	+0.67
ACSS-Ali.	+0.57	+7.99
SC-Align	+0.38	+7.66

Table 3: Average improvements of alignment-aware selection methods relative to SC-CoT across all models.

paring CoT/SC-CoT baselines against the ACSS family of selectors and SC-Align. We explore whether the proposed Alignment Score is meaningfully related to task performance, and whether it can be exploited as a test-time signal for chain selection. We study this question empirically using the ACSS family of selectors (which rank chains by alignment errors or the Alignment Score) and the SC-Align strategy (which orders chains by Alignment Score based on majority voting). Additional experiments on more recent models can be found in Appendix E.

Correlation with accuracy. Table 2 reveals three consistent empirical observations: (i) Regarding ACSS-TS and ACSS-RR, which select chains by minimizing specific alignment errors, these methods clearly improve both Accuracy and Alignment Score over the CoT baseline in most cases, but usually do not beat SC-CoT in Accuracy; (ii) when we use the Alignment Score itself as the selection signal (ACSS-Ali.), several models, including GPT-4o-mini, GPT-4o, and Qwen2.5-3B, achieve higher accuracy than SC-CoT while also enjoying substantial gains in Alignment Score; (iii) this effect is pronounced for reasoning models, where base CoT performance is strong, so SC-CoT yields diminishing or negative gains, matching studies where CoT adds little on strong models (Wang et al., 2025; Liu et al., 2025), whereas ACSS-Ali. still surpasses SC-CoT in accuracy. To make this coupling explicit, Table 3 summarizes the average improvements over SC-CoT: alignment-aware methods that directly optimize Alignment Score (ACSS-Ali. and SC-Align) deliver the largest gains in Alignment Score while improving Accuracy. These observations indicate an empirical coupling between Alignment Score and answer correctness.

High-accuracy chains are more aligned. To test whether high-accuracy chains are *intrinsically* more aligned, we further examine SC-Align, which first performs majority voting as in SC-CoT and then resolves ties by selecting the chain with the highest Alignment Score. Consistent with our theoretical view, Table 2 shows that SC-Align closely matches SC-CoT in Accuracy across all models, while achieving markedly higher Alignment Scores, often by a large margin. This pattern suggests that chains with higher probability of being correct naturally exhibit higher semantic alignment.

Aspect	Method	Score (mean±std)	Pairwise
NLI Model: roberta-large-mnli			
Coherence	SC-CoT	8.54(±0.78)	12.0% win
	SC-Align	**8.69**(±0.78)	**28.5% win**
Readability	SC-CoT	8.23(±0.58)	22.5% win
	SC-Align	**8.45**(±0.59)	**45.5% win**
NLI Model: deberta-v3-large-mnli			
Coherence	SC-CoT	8.49(±0.84)	20.0% win
	SC-Align	**8.54**(±0.92)	**28.5% win**
Readability	SC-CoT	8.30(±0.61)	26.5% win
	SC-Align	**8.44**(±0.64)	**43.0% win**
NLI Model: bart-large-mnli			
Coherence	SC-CoT	8.55(±0.84)	14.0% win
	SC-Align	**8.63**(±0.95)	**26.5% win**
Readability	SC-CoT	8.22(±0.59)	19.5% win
	SC-Align	**8.50**(±0.61)	**49.0% win**

Table 4: LLM-based evaluation of reasoning quality (SC-CoT vs. SC-Align) under three distinct NLI backbones. Scores are on a 1–10 scale.

LLM-based evaluation with ablations. Table 4 reports LLM-based evaluation for SC-CoT and SC-Align under three distinct NLI backbones (Liu et al., 2019; He et al., 2023; Lewis et al., 2020). Across all NLI backbones, SC-Align attains higher mean scores on both coherence and readability and is preferred more frequently in pairwise comparisons. We additionally conduct an extensive grid ablation (see Appendix F) and observe the same pattern. These consistent trends indicate that: Alignment Score is insensitive to the specific NLI backbone and behaves as a stable and reliable signal that tracks human preferences. We also provide a qualitative case study about the evaluation results, which can be found in Appendix G.

5 Discussion: Applicability Beyond CoT

Although we study Alignment Score under standard CoT prompting, the metric is not tied to CoT itself. Rather, CoT serves as a controlled setting in which intermediate reasoning states are explicitly segmented into ordered steps. More generally, Alignment Score applies to structured reasoning trajectories whose intermediate states can be decomposed into comparable units and analyzed through their pairwise semantic relations. This broader view is also motivated by recent work showing that intermediate trajectories in reasoning and agent-based systems may themselves be vulnerable to failure and manipulation (Hu et al., 2026a,b).

Generalizability to other reasoning frameworks. Here, we discuss the applicability of Alignment Score to representative reasoning frameworks. In Reflexion (Shinn et al., 2023), the initial answer, critique, and revised answer can be treated as staged segments. In ReAct (Yao et al., 2023), each thought–action–observation cycle can be viewed as a structured step, allowing consistency to be assessed across reasoning and interaction. More broadly, the framework may be applicable to agent-style reasoning settings with explicit intermediate trajectories, although richer cases such as branching or partially ordered reasoning would require additional alignment design.

6 Conclusion

In this paper, we take CoT as a concrete entry point for measuring *structured reasoning alignment* in large language models. We introduce the *Alignment Score*, a lightweight measure that aligns generated chains with reference demonstrations in a one-to-one manner, and show across diverse settings that it is coupled with task accuracy and overall quality. As CoT demonstrations (reference chains) become more logically coherent, readable, correct, and better aligned with human preferences, the Alignment Score increases and accuracy improves. Moreover, we empirically show that optimizing the Alignment Score will lead to increased readability and logical coherence. In this sense, CoT moves from being a prompting heuristic to serving as a quantitative lens on reasoning quality and a diagnostic signal.

Limitations

This work has several limitations:

- The proposed Alignment Score is designed for *step-structured* reasoning traces with clearly separated statements. Many real-world explanations are written as free-form paragraphs without explicit step boundaries and therefore require additional processing. In such settings, our entropy-based alignment framework is not directly applicable. Extending Alignment Score to less structured forms of reasoning remains a challenge.
- Our experiments are conducted on multiple-choice QA benchmarks with a single correct option. These settings provide a clean testbed, but they do not cover open-ended tasks such as code generation. How well Alignment Score and our alignment-aware methods generalize to these broader scenarios remains to be tested.
- In Section 4.3, we empirically show that reasoning chains with more hops are more susceptible to certain alignment errors. However, we do not yet provide a theoretical account or formal characterization of why this degradation emerges.

Ethical Considerations

Our work is primarily methodological, but it still raises several ethical considerations about evaluation bias, interpretation, and potential deployment.

Data and privacy. This work uses public multiple-choice QA benchmarks and AI-assisted reference-chain generation, and therefore does not involve personally identifiable information. We checked data and case reporting to ensure that no information requiring anonymization is included.

Potential bias in LLM-based evaluation. Because Alignment Score is defined relative to curated reference chains, it may prefer some reasoning styles over other equally valid ones. In addition, our LLM-as-a-judge pipeline may introduce stylistic bias when evaluating readability and coherence.

Interpretation and deployment. A high Alignment Score must not be taken as a guarantee of truthfulness, accuracy, or safety. We regard the Alignment Score as a diagnostic indicator, not as a sufficient condition on its own for use in high-risk domains, including medical applications.

Acknowledgments

This work is partially funded by the European Union (under grant agreement ID 101212818). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Health and Digital Executive Agency (HADEA). Neither the European Union nor the granting authority can be held responsible for them. This work is partially supported by Innovate UK through AI-PASSPORT under Grant 10126404. This work was awarded a grant by the AI Security Institute (AISI) via the Alignment Project (Rare-Event Estimation in Large Language Models via Subset Simulation) and funded by EPSRC. Yi's contribution is partially supported through the Royal Society international exchanges programme and in part by the Engineering and Physical Sciences Research Council, through funding from RAi UK [EP/Y009800/1].

References

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 2024. MT-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7421–7454, Bangkok, Thailand. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2024. Overreasoning and redundant calculation of large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–169, St. Julian's, Malta. Association for Computational Linguistics.

François Chollet. 2019. On the measure of intelligence. *Preprint*, arXiv:1911.01547.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Nelson Cowan. 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1):87–114.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Jinwei Hu, Yi Dong, Shuang Ao, Zhuoyun Li, Boxuan Wang, Lokesh Singh, Guangliang Cheng, Sarvapali D. Ramchurn, and Xiaowei Huang. 2025. Stop reducing responsibility in llm-powered multi-agent systems to local alignment. *Preprint*, arXiv:2510.14008.

Jinwei Hu, Yi Dong, Youcheng Sun, and Xiaowei Huang. 2026a. Tapas are free! training-free adaptation of programmatic agents via llm-guided program synthesis in dynamic environments. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(35):29477–29485.

Jinwei Hu, Xinmiao Huang, Youcheng Sun, Yi Dong, and Xiaowei Huang. 2026b. Lying with truths: Open-channel multi-agent collusion for belief manipulation via generative montage. *Preprint*, arXiv:2601.01685.

Hui Huang, Xingyuan Bu, Hongli Zhou, Yingqi Qu, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. 2025. An empirical study of LLM-as-a-judge for LLM evaluation: Fine-tuned judge model is not a general substitute for GPT-4. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5880–5895, Vienna, Austria. Association for Computational Linguistics.

Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roee Aharoni, and Mor Geva. 2024. A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4615–4634, Bangkok, Thailand. Association for Computational Linguistics.

Joonwon Jang, Jaehee Kim, Wonbin Kweon, Seonghyeon Lee, and Hwanjo Yu. 2025. Verbosity-aware rationale reduction: Sentence-level rationale reduction for efficient and effective reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20769–20784, Vienna, Austria. Association for Computational Linguistics.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic entropy probes: Robust and cheap hallucination detection in llms. *Preprint*, arXiv:2406.15927.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

J. Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Ryan Liu, Jiayi Geng, Addison J. Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L. Griffiths. 2025. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. *Preprint*, arXiv:2410.21333.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, volume 35, pages 2507–2521. Curran Associates, Inc.

Huan Ma, Jiadong Pan, Jing Liu, Yan Chen, Joey Tianyi Zhou, Guangyu Wang, Qinghua Hu, Hua Wu, Changqing Zhang, and Haifeng Wang. 2025. Semantic energy: Detecting llm hallucination beyond entropy. *Preprint*, arXiv:2508.14496.

Minh-Vuong Nguyen, Linhao Luo, Fatemeh Shiri, Dinh Phung, Yuan-Fang Li, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Direct evaluation of chain-of-thought in multi-hop reasoning with knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2862–2883, Bangkok, Thailand. Association for Computational Linguistics.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. In *Advances in Neural Information Processing Systems*, volume 37, pages 8901–8929. Curran Associates, Inc.

OpenAI. 2023. Gpt-3.5 turbo. OpenAI API Documentation. Accessed: 2025-12-26.

OpenAI. 2024a. Gpt-4o. OpenAI API Documentation. Accessed: 2025-12-26.

OpenAI. 2024b. Gpt-4o mini. OpenAI API documentation. Accessed: 2025-12-26.

OpenAI. 2024c. o1. OpenAI API Documentation. Accessed: 2025-12-26.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 431 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*. Featured Certification.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Technology Innovation Institute. 2023. Falcon-7B-Instruct. Technology Innovation Institute Falcon LLM Model Page. Accessed: 2025-12-26.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Robert Vacareanu, Anurag Pratik, Evangelia Spiliopoulou, Zheng Qi, Giovanni Paolini, Neha Anna John, Jie Ma, Yassine Benajiba, and Miguel Ballesteros. 2024. General purpose verification for chain of thought prompting. *Preprint*, arXiv:2405.00204.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. 2025. Benchmarking uncertainty quantification methods for large language models with LM-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248.

Weiqin Wang, Yile Wang, and Hui Huang. 2025. Ranked voting based self-consistency of large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14410–14426, Vienna, Austria. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

Meilin Zhu, Gaojie Jin, Xiaowei Huang, and Lijun Zhang. 2025. Shapley uncertainty in natural language generation. *Preprint*, arXiv:2507.21406.

A CoT Prompting Template

For all models, we use a two-part prompt consisting of a system message and a user message. Below we list the exact templates and hyperparameter settings used in our experiments.

A.1 Base CoT Prompt with Reference

System message

```
You are a careful reasoning assistant.
Output ONLY one <json>{...}</json> block, no
    extra text.
```

User message

```
Here is a worked example of multiple-choice
    reasoning with a concise chain.

Example Question: {RQ}
Example Context: {RCTX}
Example Options: (A) {ROPT_A} (B) {ROPT_B} (C)
     {ROPT_C} (D) {ROPT_D}
Example Answer: {RANS}
Example Chain (strict JSON):
<json>{"because_steps": ["STEP_1", ..., "
    STEP_L"],
      "therefore": "FINAL_SENTENCE",
      "answer": "{RANS}"}</json>

Now solve the NEW problem in the SAME STYLE
    and EXACTLY the SAME NUMBER OF STEPS.
- Produce EXACTLY {L_steps} steps in "
    because_steps".
- Keep each step to ONE short sentence; keep
    entities consistent; avoid hedging.
- Output ONLY one
  <json>{"because_steps":["..."],"therefore
      ":"...","answer":"A"}</json> block. No
      extra text.
- The field "answer" MUST be one of A, B, C,
    or D and MUST be a single letter (e.g., "
    A").

Question: {TQ}
Context: {TCTX}
Options: (A) {TOPT_A} (B) {TOPT_B} (C) {
    TOPT_C} (D) {TOPT_D}
```

Here, `{RQ,RCTX,ROPT_*,RANS}` come from the reference chain, `{TQ,TCTX,TOPT_*}` from the target QA instance, and `{L_steps}` is the hop number $L \in \{1, 2, 3, 4\}$.

A.2 Hyperparameter Settings

OpenAI chat models. Unless otherwise specified, GPT-4o and GPT-4o-mini are called with temperature $= 0.4$, top-$p = 0.9$, n=1, and a maximum generation length of 640 tokens. For multi-sample settings (SC-CoT, SC-Align with multiple candidate chains), we increase the temperature to 0.7 while keeping top-$p = 0.9$ and the same length.

Metric	Percentile q	Threshold
TS	5	$\tau_{\text{TS}} = 0.425$
TS	10	$\tau_{\text{TS}} = 0.563$
TS	15	$\tau_{\text{TS}} = 0.712$
RR	70	$\tau_{\text{RR}} = 0.794$
RR	75	$\tau_{\text{RR}} = 0.816$
RR	80	$\tau_{\text{RR}} = 0.858$

Table 5: Threshold summary for TS and RR using percentile-based calibration.

Open-source models. Open-source LLMs are queried via the HuggingFace `text-generation` pipeline with `do_sample=true`, using the same temperature and top-p values as their OpenAI counterparts in each setting, and the same maximum length (640 tokens).

Repetitions and reporting. Each configuration (model, hop, selection strategy) is repeated over 5 random seeds. We report the mean and standard deviation of the Accuracy, the Alignment Score and LLM-based evaluation scores across these runs.

B Threshold Selection for τ_{TS} and τ_{RR}

We choose the TS/RR thresholds via percentile calibration to avoid hand-tuning absolute cosine cutoffs, which can vary with the embedding model and prompting format. As detailed in Table 5, for thematic shift (TS), we target *unusually low* semantic continuity between adjacent steps; thus τ_{TS} is set from the *lower tail* (a small percentile) of adjacent-step similarities $\{\cos(e_{B_i}, e_{B_{i+1}})\}$. For redundant reasoning (RR), we target *unusually high* semantic overlap with earlier steps; thus τ_{RR} is set from the *upper tail* (a high percentile) of $\{\max_{j<i} \cos(e_{B_i}, e_{B_j})\}$. Unless otherwise stated, we instantiate these percentiles as q_{TS}=10 and q_{RR}=75, yielding $\tau_{\text{TS}} = Q_{10}(\{\cos(e_{B_i}, e_{B_{i+1}})\})$ and $\tau_{\text{RR}} = Q_{75}(\{\max_{j<i} \cos(e_{B_i}, e_{B_j})\})$, where $Q_q(\cdot)$ denotes the q-th percentile. We compute these percentiles on a development split constructed from 256 sets of 4-hop reasoning chains generated by LLaMA-2-13B-Chat, which provides a reference distribution for calibrating semantic continuity and redundancy.

C LLM-based Evaluation

C.1 Prompt Template

We use an LLM-as-judge protocol to compare the baseline chain with the SC-Align chain for the same multiple-choice question. For each matched pair $(\mathrm{Hop}, \mathrm{Idx})$, we construct two textual chains and randomly decide which one is shown as "Chain 1" and which as "Chain 2". The judge model evaluates both chains along two dimensions:

- **Logical completeness**: whether the chain covers the key reasoning steps needed to justify the answer and maintains coherent logic.
- **Readability**: whether the chain is easy to understand and free of confusing repetition.

The judge assigns integer scores in $[1, 10]$ for each dimension and chain, decides which chain is better (or a tie) on each dimension, and provides a brief textual explanation. The order of chains is stored in the metadata so that scores can be mapped back to baseline vs. SC-Align.

System message (judge)

```
You are a strict but fair evaluator of
reasoning quality.
```

User message (judge)

```
You are an expert evaluator of reasoning chains.
You will be given:
- A multiple-choice question and its options.
- Two reasoning chains that both try to answer this question:
  Chain 1 is produced by **{CHAIN1_NAME}**.
  Chain 2 is produced by **{CHAIN2_NAME}**.
You must judge each chain along two dimensions:
1. Logical completeness: Does the chain cover the key reasoning steps needed
   to justify the answer? Is the causal logic coherent and sufficiently detailed?
2. Readability: Is the chain easy to understand, well-structured, and free of
   confusing repetition?
For EACH dimension, assign a score from 1 (very poor) to 10 (excellent) to BOTH
chains. Then decide which chain is better on that dimension (or "tie" if they
are comparable).
Return your judgment as a JSON object with the following fields ONLY:
{
  "chain1_logic": <integer 1-10>,
  "chain2_logic": <integer 1-10>,
  "chain1_readability": <integer 1-10>,
  "chain2_readability": <integer 1-10>,
  "better_logic": "chain1" | "chain2" | "tie",
  "better_readability": "chain1" | "chain2" | "tie",
  "explanation": "<brief explanation in 1-3 sentences>"
}
Make sure the output is valid JSON (no comments, no trailing commas).
Question:
{QUESTION}
Options:
{OPTION_KEY}. {OPTION_TEXT}
...
--- Chain 1 ---
{CHAIN1_TEXT}
--- Chain 2 ---
{CHAIN2_TEXT}
```

D Construction of Reference Chains

To obtain high-quality reference chains for each hop L ∈ 1, 2, 3, 4, we apply an AI-assisted curation pipeline. Each reference chain is an exemplar with exactly L intermediate steps plus a final conclusion, stored in JSON format.

Candidate generation. We start from a pool of multiple-choice questions with gold labels. For each question and hop L, we prompt a strong LLM (e.g., GPT-4o) to generate K candidate chains that (i) predict an explicit answer option and (ii) output a structured JSON object. We discard all candidate reasoning chains whose predicted answer does not correctly match with the gold answer.

Quality scoring via LLM-as-judge. For the remaining candidates of each question, we invoke a separate "judge" LLM to rate each chain along three dimensions: *readability*, *logical coherence*, and *step granularity* (whether each step corresponds to a single, non-trivial reasoning move), following the method shown in Appendix C. The judge returns integer scores (e.g., 1–5) for each dimension and a short textual justification.

Chain selection. For each question, the judge scores are aggregated into a single ranking score (with higher weight on logical coherence). We automatically select the top-scoring chain as the reference chain for hop L. Chains containing obvious issues (duplicated steps, self-contradictions, generic meta-commentary such as "I will now think step by step", placeholder tokens, etc.) are rejected by simple rule-based filters.

Human sanity check. To validate the pipeline, we randomly sample a small subset (5 percent) of the automatically selected reference chains for man-

Method	Open-Source Models Qwen3-4B-Instruct	Closed-Source Models GPT-5-mini	GPT-5
	Accuracy (mean ± std, %)		
CoT	89.30 (±2.46)	95.67 (±1.21)	94.47 (±0.89)
SC-CoT	**90.86** (±2.85)	95.62 (±0.37)	**96.81** (±2.18)
ACSS-TS	89.35 (±2.82)	95.55 (±1.72)	95.38 (±1.58)
ACSS-RR	89.22 (±4.94)	95.43 (±1.20)	95.70 (±2.38)
ACSS-Ali.	**91.13** (±2.72)	**96.41** (±0.82)	95.84 (±2.67)
SC-Align	90.38 (±2.56)	**95.78** (±2.23)	**96.72** (±1.80)
	Alignment Score (mean ± std)		
CoT	78.37 (±13.07)	78.57 (±12.95)	78.57 (±13.31)
SC-CoT	78.48 (±13.41)	79.00 (±13.13)	78.84 (±12.90)
ACSS-TS	78.30 (±13.10)	79.33 (±12.26)	79.45 (±13.48)
ACSS-RR	78.39 (±13.37)	78.75 (±13.37)	79.07 (±12.87)
ACSS-Ali.	**84.23** (±11.25)	**88.02** (±8.12)	**87.57** (±8.22)
SC-Align	**84.94** (±10.25)	**87.74** (±8.09)	**88.00** (±8.10)

Table 6: Additional experiments under the same setup as the main paper. **Red** numbers indicate the best result, and **blue** numbers indicate the second best.

ual inspection. This sanity check confirms that the curated chains are logically sound and stylistically consistent, while keeping the annotation cost low.

E Additional Experiments

To further test the generality of our findings, we evaluate three additional models, namely Qwen3-4B-Instruct, GPT-5-mini, and GPT-5, under the same setup as in the main paper. As shown in Table 6, the overall trend remains consistent: improvements in Alignment Score are closely associated with gains in accuracy.

F Grid Ablation on LLM Judge Models

As shown in Table 7, we conduct a *grid ablation* over both the **LLM judge** and the **NLI backbone**. Specifically, we repeat the SC-CoT vs. SC-Align comparison under three LLM judges (GPT-4o, GPT-4.1, and GPT-3.5-turbo) and three NLI backbones (RoBERTa-large-MNLI, DeBERTa-v3-large-MNLI, and BART-large-MNLI), forming a 3×3 evaluation grid. Across all nine configurations, SC-Align consistently achieves higher mean ratings in both *coherence* and *readability*, and is preferred in pairwise comparisons by a clear margin. The persistence of this advantage across diverse judges and backbone choices suggests that the observed gains reflect a robust improvement in reasoning-chain quality, rather than an artifact of any particular judge or NLI backbone.

Aspect	Method	Score (mean±std)	Pairwise
NLI Backbone: RoBERTa-large-MNLI			
Judge Model: GPT-4o			
Coherence	SC-CoT	8.85(±0.71)	16.1% win
	SC-Align	**8.91**(±0.76)	**23.9% win**
Readability	SC-CoT	8.65(±0.56)	14.5% win
	SC-Align	**8.87**(±0.49)	**37.3% win**
Judge Model: GPT-4.1			
Coherence	SC-CoT	8.55(±0.95)	35.7% win
	SC-Align	**8.70**(±0.99)	**53.3% win**
Readability	SC-CoT	8.80(±0.75)	27.5% win
	SC-Align	**9.07**(±0.75)	**51.8% win**
Judge Model: GPT-3.5-turbo			
Coherence	SC-CoT	8.37(±0.58)	23.1% win
	SC-Align	**8.60**(±0.61)	**44.7% win**
Readability	SC-CoT	8.06(±0.73)	28.6% win
	SC-Align	**8.32**(±0.54)	**52.5% win**
NLI Backbone: DeBERTa-v3-large-MNLI			
Judge Model: GPT-4o			
Coherence	SC-CoT	8.85(±0.75)	14.1% win
	SC-Align	**8.95**(±0.69)	**23.5% win**
Readability	SC-CoT	8.69(±0.58)	12.9% win
	SC-Align	**8.93**(±0.48)	**34.5% win**
Judge Model: GPT-4.1			
Coherence	SC-CoT	8.57(±0.90)	36.5% win
	SC-Align	**8.76**(±0.89)	**52.2% win**
Readability	SC-CoT	8.86(±0.76)	31.0% win
	SC-Align	**9.07**(±0.66)	**48.2% win**
Judge Model: GPT-3.5-turbo			
Coherence	SC-CoT	8.39(±0.58)	26.3% win
	SC-Align	**8.54**(±0.67)	**40.0% win**
Readability	SC-CoT	8.11(±0.69)	28.2% win
	SC-Align	**8.36**(±0.57)	**52.9% win**
NLI Backbone: BART-large-MNLI			
Judge Model: GPT-4o			
Coherence	SC-CoT	8.83(±0.66)	14.9% win
	SC-Align	**8.92**(±0.68)	**24.3% win**
Readability	SC-CoT	8.70(±0.53)	14.9% win
	SC-Align	**8.91**(±0.49)	**36.1% win**
Judge Model: GPT-4.1			
Coherence	SC-CoT	8.55(±0.97)	39.2% win
	SC-Align	**8.69**(±0.90)	**49.0% win**
Readability	SC-CoT	8.78(±0.71)	31.0% win
	SC-Align	**9.00**(±0.74)	**51.4% win**
Judge Model: GPT-3.5-turbo			
Coherence	SC-CoT	8.35(±0.58)	28.6% win
	SC-Align	**8.53**(±0.66)	**45.5% win**
Readability	SC-CoT	8.10(±0.70)	30.2% win
	SC-Align	**8.36**(±0.56)	**54.5% win**

Table 7: LLM-based evaluation of reasoning quality (SC-CoT vs. SC-Align) under three different LLM judges, across three NLI backbones used for Alignment Score calculation (RoBERTa-large-MNLI, DeBERTa-v3-large-MNLI, and BART-large-MNLI). Scores are on a 1–10 scale. **Bold** numbers indicate the higher values.

G Qualitative Case Study of Reasoning Alignment Comparison

To complement the quantitative results, we conduct a qualitative inspection of the LLM-evaluation results and select representative SC-CoT vs. SC-Align pairs where the baseline chain exhibits a lower reasoning alignment and SC-Align exhibits a higher reasoning alignment according to our metric. In all cases, low-alignment baseline chains tend to (i) leave key relations implicit (missing "bridge" statements), (ii) repeat semantically similar points without making step-wise progress, or (iii) contain small local incoherence that reduces readability even when the conclusion is plausible. In contrast, SC-Align is often preferred because it states the intended causal/semantic links more explicitly and presents the steps in a more coherent and readable form. In the following, we summarize the results of the LLM-based evaluation and provide a qualitative analysis. We also include a failure case to illustrate a common error mode.

Case 1: Making implicit entity roles and properties explicit Here, the judge prefers SC-Align because it more explicitly distinguishes key entities and clarifies their roles and properties, resulting in a rationale that is both easier to follow and more logically coherent.

SC-Align more explicitly distinguishes the two groups and improves logic and readability

Question: Some researchers believe that beta-carotene, which is found in carrots, may help humans live longer lives. A few scientists designed an experiment in which one-hundred people would be asked to take a double dose of beta-carotene daily for two years and another hundred people were given a placebo, a sugar pill, for the same amount of time. After the two years, the scientists planned to collect the data and draw a conclusion. This is not a good example of a scientific study and was not done because there is no control group. Is that correct?

Baseline (SC-CoT; lower Alignment Score):

Step 1: A control group is essential for comparing results against the experimental group.

Step 2: The placebo group serves as the control group in this study.

Step 3: Without a control group, it is difficult to determine the effect of beta-carotene.

Step 4: This study does have a control group, so that is not the issue.

Conclusion: The statement that there is no control group is incorrect.

SC-Align (higher Alignment Score):

Step 1: The study includes one group taking beta-carotene and another group taking a placebo.

Step 2: The group taking beta-carotene is the experimental group.

Step 3: The placebo group serves as the control group.

Step 4: Because the study includes a placebo control group, the absence of a control group is not the flaw in the design.

Conclusion: The statement that there is no control group is incorrect.

LLM Judge Scores:

Logic: Baseline vs SC-Align (better: SC-Align)

Readability: Baseline vs SC-Align (better: SC-Align)

LLM Judge Explanation: SC-Align is preferred because it more explicitly distinguishes the two groups and states their roles more directly, making the reasoning easier to follow and logically clearer.

Case 2: Reducing redundancy while preserving the same reasoning trajectory. In this pair, both chains are logically correct, but SC-Align is preferred because it avoids minor repetition and uses more concise phrasing, improving readability without changing the underlying logic.

Same chemistry but less repetition: SC-Align keeps the steps but removes redundancy

Question: Robert put two drops of an indicator into vinegar, and the color changed. Then he added ammonia solution until the color disappeared. What process occurred?

Baseline (SC-CoT; lower Alignment Score):

Step 1: The indicator changes color in response to acidity or basicity.

Step 2: Vinegar is acidic, turning the indicator red.

Step 3: Adding ammonia, which is basic, neutralizes the acid in vinegar.

Step 4: The disappearance of red color indicates a neutralization reaction occurred.

Conclusion: The process that occurred is neutralization.

SC-Align (higher Alignment Score):

Step 1: The indicator changes color in response to pH changes.

Step 2: Vinegar is acidic and turns the indicator red.

Step 3: Ammonia is a base that can neutralize acids.

Step 4: When the color disappears, it indicates a neutralization reaction occurred.

Conclusion: The process that occurred is neutralization.

LLM Judge Scores:

Logic: Baseline vs SC-Align (better: tie)

Readability: Baseline vs SC-Align (better: SC-Align)

LLM Judge Explanation: Both chains logically explain the neutralization process. SC-Align is more readable due to its concise phrasing and lack of redundancy.

Case 3: Readability gains under the same logical content. This example illustrates a frequent pattern in our logs: the judge considers the logic comparable, but SC-Align is preferred because it is phrased more cleanly and reads more naturally.

Equivalent causal story but clearer phrasing: readability improves with logic unchanged (logic tie)

Question: Which statement best explains why a tree branch floats on water?

Baseline (SC-CoT; lower Alignment Score):

Step 1: Wood has air pockets that make it less dense than water.

Step 2: Because it is less dense, it will float on water.

Step 3: If an object is less dense than water, it will float.

Step 4: Therefore, wood is buoyant.

Conclusion: This means wood is buoyant.

SC-Align (higher Alignment Score):

Step 1: Wood contains air-filled spaces that lower its overall density.

Step 2: Lower density than water allows an object to float.

Step 3: A floating object demonstrates buoyancy.

Step 4: Thus, the branch floats because wood is buoyant.

Conclusion: This means wood is buoyant.

LLM Judge Scores:

Logic: Baseline vs SC-Align (better: tie)

Readability: Baseline vs SC-Align (better: SC-Align)

LLM Judge Explanation: Both chains provide a complete and coherent reasoning process. However, SC-Align is slightly more readable due to its clearer phrasing and structure.

Case 4 (Failure case): SC-Align introduces an extra assumption and is penalized. Finally, we include a counterexample where the baseline is preferred. This highlights a typical failure mode: SC-Align may add an unnecessary or less-supported assumption, which can reduce logical precision and hurt readability.

Failure mode: added assumption reduces alignment (baseline preferred by the judge)

Question: Some animals live in a grassland. A fire burns through the grassland. Which type of animal will best compete for resources after the fire?

Baseline (SC-CoT; lower Alignment Score):

Step 1: The fire destroys above-ground vegetation.

Step 2: Animals that rely on plants for shelter will struggle after the fire.

Step 3: Animals that live underground can avoid the immediate effects of the fire.

Step 4: Those animals can access resources that are less affected by the fire.

Conclusion: An animal that lives under the ground will best compete for resources after the fire.

SC-Align (higher Alignment Score):

Step 1: The fire destroys above-ground vegetation.

Step 2: Animals that rely on plants for shelter will struggle after the fire.

Step 3: Animals that live underground can avoid the immediate effects of the fire.

Step 4: Animals that build nests on the ground may lose their nests and face resource competition.

Conclusion: An animal that lives under the ground will best compete for resources after the fire.

LLM Judge Scores:

Logic: Baseline vs SC-Align (better: baseline)

Readability: Baseline vs SC-Align (better: baseline)

LLM Judge Explanation: Baseline is more logically complete because it directly connects underground living to accessing resources that remain available after the fire. SC-Align introduces extra details about nests that are not necessary for the conclusion, which slightly reduces clarity.

Summary. Overall, these case studies align with the intended behavior of Alignment Score: lower-alignment chains are often less coherent due to missing links, redundancy, or added assumptions, while higher-alignment SC-Align chains tend to be more coherent and more readable. The failure case further motivates using alignment-based diagnostics to detect when a rewrite adds unnecessary content or weakens the original reasoning structure.