Evaluating Step-by-step Reasoning Traces: A Survey

Jinu Lee and **Julia Hockenmaier**
University of Illinois Urbana-Champaign
{jinulee2, juliahmr}@illinois.edu

Abstract

Step-by-step reasoning is widely used to enhance the reasoning ability of large language models (LLMs) in complex problems. Evaluating the quality of reasoning traces is crucial for understanding and improving LLM reasoning. However, existing evaluation practices are highly inconsistent, resulting in fragmented progress across evaluator design and benchmark development. To address this gap, this survey provides a comprehensive overview of step-by-step reasoning evaluation, proposing a taxonomy of evaluation criteria with four top-level categories (factuality, validity, coherence, and utility). Based on the taxonomy, we review different datasets, evaluator implementations, and recent findings, leading to promising directions for future research.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in reasoning on complex problems, such as logic, math, and science. At the core of this versatility lies **step-by-step reasoning** (Wei et al., 2022; Kojima et al., 2022), where the LLM generates an intermediate reasoning trace before presenting the final answer.

The reasoning ability of LLMs is often measured in terms of *answer accuracy*, *i.e.*, finding the correct answer for a complex reasoning problem (Cobbe et al., 2021; Zhong et al., 2021). However, answer accuracy is generally insufficient for measuring LLMs' reasoning ability, as the correct answer does not imply the correctness of the preceding reasoning trace (Lanham et al., 2023; Mirzadeh et al., 2024; Paul et al., 2024). Furthermore, assessing the quality of the reasoning trace can directly lead to better reasoning ability of LLMs by verifier-guided search (Wang et al., 2023b; Yao et al., 2023; Hao et al., 2024) and reinforcement learning (Lu et al., 2024; Cui et al., 2025).

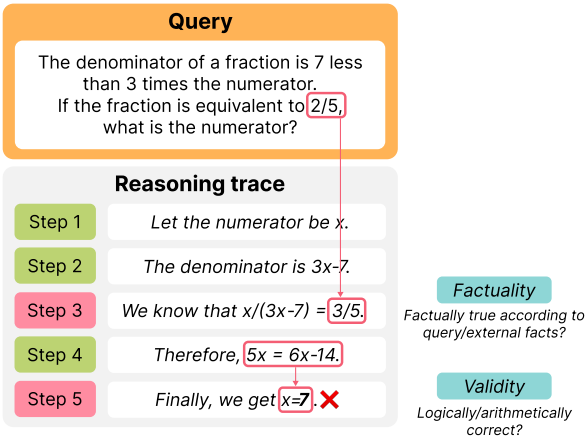

Figure 1: Illustrative example of reasoning trace evaluation.

Consequently, reasoning trace evaluation is an active research topic, with numerous new evaluators and datasets continuously being proposed. However, this rapid growth has led to a proliferation of evaluators and datasets without establishing a consensus on the **criteria** (*what to evaluate*). In this survey, we aim to provide a systematic review of existing step-by-step reasoning evaluation criteria, which will serve as a foundation for implementing evaluators.

Implementing an evaluator also introduces several practical decision choices. Different architectures involve trade-offs between computational cost and expected performance, while non-architectural factors like training data and data format also play a significant role. This survey seeks to categorize and compare various evaluator implementations, highlighting key trade-offs and revealing additional dimensions that merit consideration.

The key contributions of this survey are:

- Defining a clear, universal taxonomy of step-by-step evaluation **criteria** (§3).
- Surveying existing **datasets** and **evaluators** for step-by-step reasoning evaluation based

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 1789–1814
November 4-9, 2025

on their implementations, across diverse reasoning tasks and criteria (§4-§5).

- Identifying recent findings and promising directions for trace evaluation (§6-§7).

2 Background

2.1 Step-by-step reasoning

Step-by-step reasoning is where LLMs generate a series of intermediate natural language steps ("thoughts") before outputting the final answer (Wei et al., 2022). Each instance consists of two parts: a **query** and a **reasoning trace**, and the **final answer** as a part of the reasoning trace. Upon seeing the query (user input), the LLM autoregressively generates its solution as a reasoning trace. Finally, a trace should include a **final answer** for the query, which can be compared to the ground truth. See Appendix A for details on different reasoning tasks.

2.2 Evaluation

Reasoning trace **evaluators** assess the quality of the reasoning trace and assign a score, reflecting whether it is good or not based on the criterion. Evaluators can be intrinsic metrics like uncertainty to models specialized for reasoning trace evaluation; see Section 5 for different types of evaluators.

2.3 Meta-evaluation

How can we *evaluate these evaluators (meta-evaluation)*? Two common directions apply: (1) using meta-evaluation benchmarks with step-wise labels, or (2) measuring the improvement in the downstream task performance (Figure 2).

2.3.1 Meta-evaluation Benchmarks

Meta-evaluation benchmarks contains labels indicating a step's quality based on the predefined criteria. In this setting, the evaluator's performance is measured by the classification accuracy of these labels. These benchmarks offer a fine-grained view of which criteria the evaluator can handle well and which cannot (Song et al., 2025). However, constructing these data often requires costly human annotation (Lightman et al., 2024; Zheng et al., 2024a) and the gains in meta-evaluation benchmark might not generalize to downstream performance (Zhang et al., 2025). Further details can be found in Appendix B.

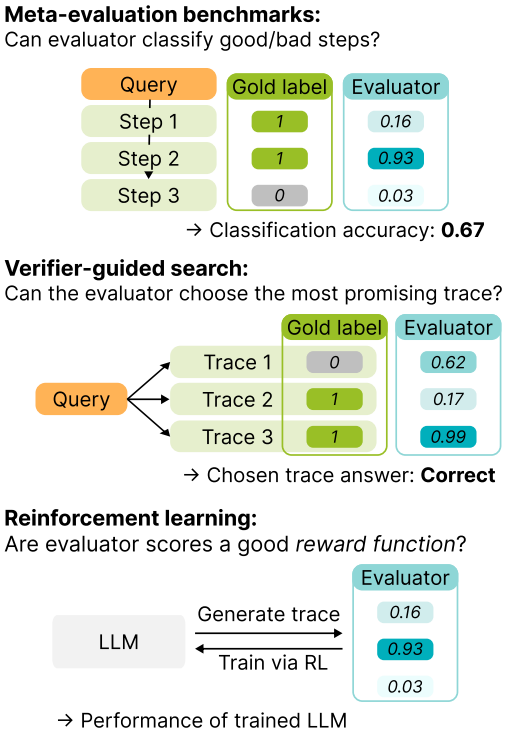

Figure 2: Illustration of three popular meta-evaluation methods: meta-evaluation benchmarks, verifier-guided search, and reinforcement learning.

2.3.2 Downstream performance improvement

As the fundamental goal of evaluators is to *improve* the reasoning ability of LLMs, the evaluator's quality can also be measured by the improvement in downstream reasoning tasks.

Verifier-guided search uses evaluator scores to find the most promising trace after exploring different paths. Popular methods include **Best-of-N decoding** (independently sampling N traces and selecting one) (Lightman et al., 2024; Zhang et al., 2024c) and **tree search** (sampling multiple candidate steps and choosing the most promising path) (Yao et al., 2023; Guan et al., 2024; Zhu et al., 2024b). The performance is often compared to majority voting without evaluators (Self-consistency; Wang et al. (2023b)), where a bigger gap indicates a better evaluator performance.

Reinforcement learning (RL) uses evaluator scores as a *reward* to further train an LLM (Uesato et al., 2022; Pan et al., 2023b; Zhang et al., 2024a). If the evaluator provides useful training rewards, the trained model will reach higher final answer accuracy. Moreover, as evaluators that are vulnerable to spurious features like length lead to *reward hacking*, successful RL also indicates the evaluator's robustness (Zhang et al., 2024a).

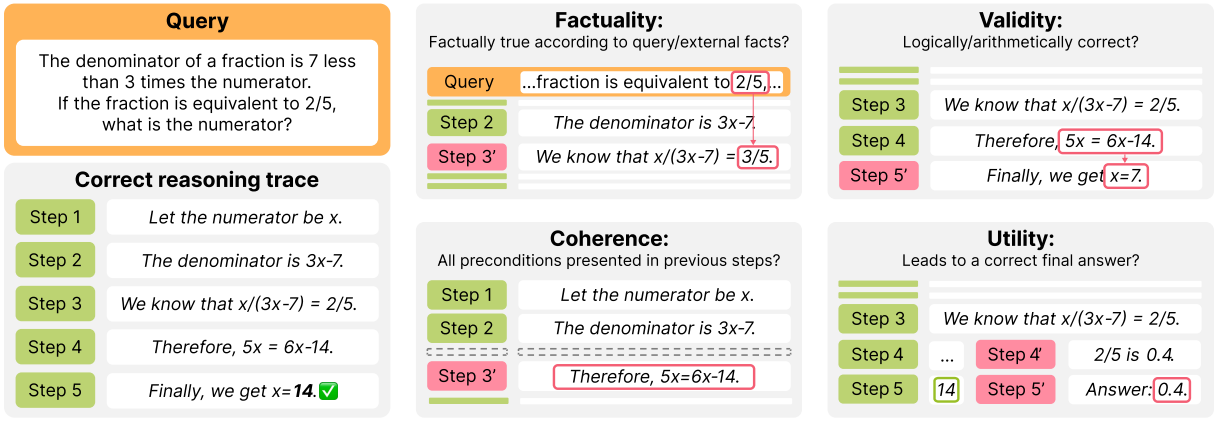

Figure 3: Illustration of the proposed categories of step-by-step reasoning evaluation criteria, *i.e.* factuality, validity, coherence, and utility. The left shows an example of a query and a reasoning trace. The other four blocks demonstrate examples that fail to satisfy the respective metric. Red filled rectangles indicate the error's location, and the outlined boxes and arrows show the cause of the error. The trace example is originally from Lightman et al. (2024).

3 Evaluation Criteria

Previous studies have proposed various criteria for evaluating step-by-step reasoning (Golovneva et al., 2023a; Lightman et al., 2024; Wang et al., 2024c; Jacovi et al., 2024), but these works failed to propose a complete taxonomy that covers diverse reasoning tasks (*e.g.*, literature in factual reasoning and math reasoning have focused on different criteria). In this section, we propose a unified taxonomy of reasoning trace evaluation criteria that spans different reasoning tasks and evaluators. We categorize them into four key dimensions: **Factuality**, **Validity**, **Coherence**, and **Utility** (Figure 3)[1].

3.1 Factuality

Factuality evaluates if the factual information can be grounded in reliable sources.

The narrower notion of factuality is **groundedness**, where the generated trace should be true according to the *query* (Lewis et al., 2020; Gao et al., 2024d). For instance, if the retrieved document explicitly mentions that *Einstein was born in 1879*, the step mentioning that he was born in *1789* is ungrounded. In less factual tasks like math, groundedness also indicates using correct numbers and constraints given in the query.

However, the reasoning process might require factual knowledge not directly mentioned in the query. This type of factuality can be referred to as **parametric knowledge**. While steps containing trivia-style facts can be readily verified by retrieval-based fact checkers (Thorne et al., 2018), verifying subtle, commonsensical knowledge remains an open challenge (Toroghi et al., 2024).

3.2 Validity

Validity evaluates if a reasoning step contains no logical errors.

The validity of a reasoning step can be defined in terms of *entailment* (Bowman et al., 2015), which is widely accepted in factual/commonsense-based reasoning (Prasad et al., 2023; Wu et al., 2024a). Under this definition, a step is considered valid if it can be directly entailed from previous steps (Tafjord et al., 2021; Dalvi et al., 2021; Saparov and He, 2023) or at least does not contradict them (Golovneva et al., 2023a; Prasad et al., 2023; Zhu et al., 2024b).

In tasks like math or logic, the more common form of validity is *correctness*, *e.g.* performing accurate calculations in arithmetic reasoning (Lightman et al., 2024; Jacovi et al., 2024; Zheng et al., 2024a) or inferring the correct logical conclusion based on the provided premises (Wu et al., 2024b; Song et al., 2025).

3.3 Coherence

Coherence measures if a reasoning step's *preconditions are satisfied* by the previous steps (Wang et al., 2023a; Lee and Hwang, 2025). For instance, if a trace includes the reasoning step *"Next, we add 42 to 16."* but the origin of the value 42 was never explained in the previous steps; this step is considered incoherent. An intuitive way to obtain an incoherent trace is randomly shuffling a coherent trace (Wang et al., 2023a; Nguyen et al., 2024), as

[1]These criteria are independent but not mutually exclusive (a step can fail to satisfy multiple criteria).

Dataset	Train	Eval	Domain	Criteria	# Trace	Human
ROSCOE (Golovneva et al., 2023b)		●	Math, Common	FVU	1.0k	●
RAGTruth† (Niu et al., 2024)	●	●	Fact	F	5.9k	●
HaluEval† (Li et al., 2023a)	●	●	Fact	F	10k	▲
Math-Shepherd (Wang et al., 2024c)	●		Math	U	440k	×
PRM800k (Lightman et al., 2024)	●	●	Math	V	75k	●
REVEAL (Jacovi et al., 2024)		●	Common	FVC	3.4k	●
MATH-Minos (Gao et al., 2024a)	●		Math	V	440k	×
SCDPO (Lu et al., 2024)	●		Math	U	30k	×
MR-GSM8k (Zeng et al., 2024a)		●	Math	V	3.0k	●
BIG-Bench-Mistake (Tyen et al., 2024)		●	Symbolic	VCU	2.2k	●
CriticBench (Lin et al., 2024)		●	Math, Common, Symbolic	VU	3.8k	×
ProcessBench (Zheng et al., 2024a)		●	Math	V	3.4k	●
MR-Ben (Zeng et al., 2024b)		●	Science, Deductive, Coding	V	6.0k	●
MR-MATH (Xia et al., 2025)		●	Math	VU	0.1k	●
PRMBench (Song et al., 2025)		●	Math	VCU	6.2k	▲
PRM-Clinic (Wang et al., 2025a)		●	Expert(Clinic)	FVC	9.7k	×
VersaPRM (Zeng et al., 2025)	●		Expert	FV	84.1k	×
BiGGenBench† (Kim et al., 2025a)		●	Math, Logic	Custom	0.1k	×

Table 1: List of evaluator training data and meta-evaluation benchmarks. † symbol indicates that the datasets include other tasks, such as summarization, instruction following, *etc*, where the **# Trace** column only counts the reasoning subset. **Train/Eval** columns denote if the dataset is used for training or meta-evaluation. **Domain** indicates what tasks are used to sample the reasoning trace. **Criteria** column shows the criteria used to annotate the data classified according to Section 3, where FVCU stands for factuality, validity, coherence, and utility, respectively. BiGGenBench (Kim et al., 2025a) applies hand-written, query-specific evaluation criteria (**Custom**). **Human** column indicates human annotation, where ● ▲ × denotes full human annotation, automatic annotation/perturbation with human verification, and full LLM-based annotation, respectively.

the premise of some steps will not appear anywhere in the previous steps (*incoherent*) even though it can be eventually deduced from the query (*valid*).

Note that coherence judgment is inherently subjective and pragmatic compared to other criteria (Jacovi et al., 2024). For instance, seemingly trivial steps like *"A part of something is present in that something"* in WorldTree V2 (Xie et al., 2020) are annotated as necessary in Dalvi et al. (2021) but not necessary in Ott et al. (2023).

3.4 Utility

Utility measures whether a reasoning step contributes to getting the correct final answer.

The narrower interpretation of utility is *progress*, or whether the step is correctly following the ground truth solution (Saparov and He, 2023; Nguyen et al., 2024). For instance, in Game of 24 (making the number 24 using 4 natural numbers and basic arithmetic operations) (Yao et al., 2023), a solution can be defined as a sequence of operations (*e.g.* $5+7=12 \rightarrow 12-6=6 \rightarrow 6\times4=24$). In this task, the utility of a step (making $5+7=12$ from 5 and 7) can be directly assessed by checking if it is a part of a correct solution.

The more general version of utility is *value function* (estimated reward). (Chen et al., 2023; Wang et al., 2024c; Setlur et al., 2024). Value function is often measured using Monte Carlo Tree Search (MCTS), where the step's value is determined by the average/maximum reward of sampled continuations. Evaluating utility as a value function offers high scalability as it only requires the gold answer for computing the reward, without any human annotation or ground-truth solutions (Wang et al., 2024c; Lai et al., 2024; Cui et al., 2025).

4 Meta-evaluation Datasets

Datasets that annotate LLM-generated reasoning traces serve as key resources for training evaluators and conducting meta-evaluations between evaluators. A summary of existing datasets is provided in Table 4.

Among these, one of the most influential is PRM800k (Lightman et al., 2024). PRM800k consists of crowdsourced tertiary validity labels (positive, negative, neutral) assigned step by step, framing reasoning trace error detection as a sequence classification problem. Its design has inspired several successors (Zeng et al., 2024a; Xia et al., 2025), setting the paradigm for subsequent reasoning trace evaluation resources.

To address different needs, several extensions have been developed. Since human annotations

Metric impl.	F	V	C	U
Rule-based	▲	▲	▲	▲
Uncertainty	●		▲	
$\mathcal{V}$-information		▲	▲	●
LLM-as-value-function				●
Cross-encoder	●	●	▲	▲
Sequence classifiers	▲	●		●
Critic models	●	●	▲	●
Generative verifiers		▲		

Table 2: Mapping between each metric implementation type to the category commonly used, where the acronym FVCU corresponds to factuality, validity, coherence, and utility, respectively. For each combination of metric and implementation, ● denotes that there are at least 3 published works, and ▲ denotes that there are 1 or 2. The full table can be found in Table 3.

are costly and difficult to scale, many works have explored automatic labeling—either by estimating utility through Monte Carlo Tree Search (MCTS) (Wang et al., 2024c; Luo et al., 2024b; Setlur et al., 2024) or by generating perturbed traces with LLMs (Lu et al., 2024; Song et al., 2025). More recent datasets further broaden the scope by enabling multi-criteria meta-evaluation (Jacovi et al., 2024; Tyen et al., 2024; Song et al., 2025) and expanding coverage beyond mathematics into diverse domains (Zeng et al., 2024b, 2025).

Additional details are provided in Appendix B.

5 Evaluator types

The goal of reasoning trace evaluators is to assess reasoning traces by assigning scores. However, choosing the right evaluator for the target criteria and task is non-trivial. For instance, there is no guarantee that evaluators designed for factuality and multi-hop question answering will seamlessly work on math reasoning problems.

In this survey, we provide a comprehensive overview of diverse reasoning trace evaluators, (Luo et al., 2024a; Wei et al., 2025). We summarize eight popular evaluator types based on the criteria they evaluate (summarized in Table 2), along with other practical strengths and weaknesses.

5.1 Rule-based matching

For tasks where the ground truth solution can be expressed as a *graph of entities*, a step corresponds to a directed edge between two entities, as in knowledge graphs for factual reasoning (Nguyen et al., 2024) or computation graphs for arithmetic problems (Li et al., 2023b). In this setting, factuality reduces to identifying the correct relation between entities, coherence to the correct ordering of steps, and utility to the existence of the step in the gold reasoning chain (Nguyen et al., 2024; Saparov and He, 2023). However, this approach does not generalize for tasks without clear symbolic representations, *e.g.*, commonsense reasoning or complex math reasoning beyond arithmetic word problems.

5.2 Intrinsic metrics

Uncertainty Uncertainty of the model can be used as an intrinsic proxy for the generated content's quality (Xiao and Wang, 2021; Zhang et al., 2023b). Qiu et al. (2024) use *token probability entropy*, defined as $\Sigma_{t \in V} p(t) \log(p(t))$ where p is the probability distribution of all tokens in vocabulary V. Farquhar et al. (2024) and Kossen et al. (2024) extend the approach by clustering semantically similar tokens and calculating the entropy for each cluster. While uncertainty-based evaluators are primarily used for factuality (Wu et al., 2024a; Farquhar et al., 2024), they have also been applied for evaluating validity (Zhu et al., 2025) or utility (Hu et al., 2024), indicating that uncertainty can be a criteria-agnostic proxy of the quality of steps.

$\mathcal{V}$-information Chen et al. (2023); Prasad et al. (2023) adopt $\mathcal{V}$-information (VI) (Hewitt et al., 2021) from information theory. Informally, VI measures if a model family $\mathcal{V}$ can generate the correct goal string g with higher probability when the target string t is given to the model. Formally, $\mathcal{V}I(t \to g) = \log p_{\mathcal{V}}(g|t) - \log p_{\mathcal{V}}(g|\phi)$ when ϕ is an empty string. When g is the final answer and t is the trace, VI becomes the difference between the answer token's probability between Chain-of-thought reasoning and zero-shot reasoning, which indicates how much information the trace provides to predicting the final answer (utility) (Chen et al., 2023). When g is a step and t is the list of previous steps, high VI means that a step is likely to follow from the context, which roughly corresponds to coherence (Prasad et al., 2023).

LLM-as-value-function RL can train LLMs to align rewards to token probabilities (relative to the *base probability* obtained from the initial model), with training objectives like DPO (Rafailov et al., 2023) and GRPO (Shao et al., 2024). For instance, when the reward is determined by the final answer correctness, the token probabilities directly correspond to utility (Mahan et al., 2024; Lai et al., 2024; Xie et al., 2024; Pang et al., 2024). Unlike sequence classifiers that lose their trace generation ability af-

Type	Name	Domains	Criteria	Note
Rule-based	DiVeRSe^E (Li et al., 2023b)	Arith	VU	Computation graph
	Direct Evaluation (Nguyen et al., 2024)	Factual	FVCU	Knowledge graph
Uncertainty	UoT (Hu et al., 2024)	Common, Expert	U	
	Entropy-based decoding (Qiu et al., 2024)	Factual	F	
	Semantic entropy probes (Farquhar et al., 2024; Kossen et al., 2024)	Factual, Common	F	
	SynCheck^E (Wu et al., 2024a)	Factual	F	
	UnCert-CoT (Zhu et al., 2025)	Code	V	
$\mathcal{V}$-information	REV (Chen et al., 2023)	Common	U	
	ReCEval^E (Prasad et al., 2023)	Common, Arith	CVU	
	EPVI (Wang et al., 2024d)	Arith, Common	U	
LLM-as-value-function	GenRM (Mahan et al., 2024)	Math, Logic, Code	U	
	V-STaR (Hosseini et al., 2024)	Arith, Code	U	
	MCTS-DPO (Xie et al., 2024)	Math, Common, Science	U	
	Step-DPO (Lai et al., 2024)	Math	U	
	Tree-PLV (He et al., 2024b)	Math, Common	U	
	Step-Controlled DPO (Lu et al., 2024)	Math	U	
	IRPO (Pang et al., 2024)	Math, Common	U	
	PRIME (Cui et al., 2025)	Math	U	
Cross-encoders	ROSCOE-LI (Golovneva et al., 2023a)	Common, Arith	FVC	Off-the-shelf
	ReCEval^E (Prasad et al., 2023)	Common, Arith	CVU	Off-the-shelf
	DiVeRSe^E (Li et al., 2023b)	Arith	VU	Off-the-shelf
	DBS (Zhu et al., 2024b)	Common, Arith, Symbolic	FVCU	Synthetic data
	SynCheck^E (Wu et al., 2024a)	Factual	F	Off-the-shelf
Sequence Classifiers	GSM8k-verifier (Cobbe et al., 2021)	Arith	U	Outcome
	PRM800K (Lightman et al., 2024)	Math	V	Process
	MATH-Minos (Gao et al., 2024a)	Math	V	Outcome/process
	Math-Shepherd (Wang et al., 2024c)	Math	U	Process
	Eurus-PRM (Yuan et al., 2024)	Math	U	Process
	PAV (Setlur et al., 2024)	Math	U	Process
	ReasonEval (Xia et al., 2025)	Math	V	Process
	Qwen-PRM (Zhang et al., 2025)	Math, Science	VU	Process
	VersaPRM (Zeng et al., 2025)	Expert	FV	Process
Critic models	Verify-CoT (Ling et al., 2023)	Math, Symbolic	V	Partial context
	Tree-of-thoughts (Yao et al., 2023)	Arith, Common	U	No fine-tune
	RAGTruth (Niu et al., 2024)	Common	F	
	CPO (Zhang et al., 2024d)	Factual, Arith	U	
	F^2-Verification (Wang et al., 2024b)	Common, Symbolic, Arith	FV	No fine-tune
	OCEAN (Wu et al., 2024c)	Factual, Common	F	
	Critic-CoT (Zheng et al., 2024b)	Math, Common, Science, Logic	U	
	AutoRace (Hao et al., 2024)	Math, Common, Logic	Custom	No fine-tune
	R-PRM (She et al., 2025)	Math	V	
	PARC (Mukherjee et al., 2025)	Math	V	No fine-tune, Partial context
	Reasoning evaluators (Kim et al., 2025b)	Math, Science, Code	V	No fine-tune
	ThinkPRM (Khalifa et al., 2025)	Math, Science, Code	V	
Generative verifiers	CLoud (Ankner et al., 2024)	Math, Logic, Code	V	
	Generative verifier (Zhang et al., 2024c)	Math, Symbolic	V	

Table 3: Evaluators for step-by-step reasoning. ^{E} denotes that the method is an **e**nsemble of different methods. **Domain** specifies the tasks the evaluator is trained/evaluated with (refer to Appendix A for more details on reasoning task). The acronym FVCU in the **Criteria** column denotes factuality, validity, coherence, and utility, respectively. For AutoRace (Hao et al., 2024), LLMs are instructed to list the criteria based on incorrect traces (Custom).

ter fine-tuning, these models retain (and improve) the ability to generate traces. However, this method requires numerous good and bad reasoning traces for training. Consequently, most existing LLM-as-value-function evaluators focus on utility, as it is easier to scale up the data by simply checking if the final answer is correct.

5.3 External evaluators

Cross-encoders Cross-encoders simultaneously encode multiple sentences using a single, small network often with millions of parameters (Devlin et al., 2019; Liu et al., 2019). They have been widely applied to solve tasks such as natural language inference (Bowman et al., 2015) and fact verification (Thorne et al., 2018), where one has to determine if the *hypothesis* can be inferred from the given *premise*. Cross-encoders trained on off-the-shelf tasks (Golovneva et al., 2023a; Zha et al., 2023; Prasad et al., 2023) or LLM-perturbed data (Zhu et al., 2024b) can be used to evaluate a reasoning step based on the query (factuality) or previous steps (validity). However, their limited language understanding ability and shorter context length restrict their performance in more complex tasks.

Sequence classifiers (Reward Models) [2] Sequence classifiers are language models with a lightweight classification head attached to the final hidden state, trained to predict a numeric score in a supervised manner (Lightman et al., 2024; Wang et al., 2024c; Setlur et al., 2024). Sequence classifiers can be further divided into (1) process (step-level) evaluator *vs.* outcome (trace-level) evaluator based on the granularity of each step (Lightman et al., 2024), and (2) *validity evaluator vs. utility evaluator* based on the source of the training data. These models achieve significant performance and efficiency (Cobbe et al., 2021; Zhang et al., 2025), but they often require costly stepwise labels for training (Lightman et al., 2024). Furthermore, they cannot generate rationales for a high or a low score, having limited explainability and leading to spurious errors (Ankner et al., 2024; She et al., 2025).

Critic models (LLM-as-a-judge) Critic models are LLMs that are trained or prompted to evaluate the reasoning traces (Zheng et al., 2023; Kim et al., 2024a; Zheng et al., 2024a; Lin et al., 2024). This approach views trace evaluation as one of many reasoning tasks, where common techniques like Chain-of-thought prompting (Huang et al., 2024a) and reinforcement learning with verifiable rewards (Chen et al., 2025a) can apply. Numerous works show that LLMs are versatile critics; they can effectively evaluate factuality, validity, coherence, and utility in diverse reasoning tasks with or without fine-tuning (Yao et al., 2023; Jacovi et al., 2024; Wu et al., 2024d; Niu et al., 2024). While conceptually simple and compatible with closed-source models, generating the rationales requires significant execution time and computation compared to other evaluator types.

Generative Verifiers This paradigm lies in the middle ground of sequence classifiers and critic models. These models first autoregressively generate the evaluation rationale as critic models do. When the generation terminates, like sequence classifiers, a small, fine-tuned head predicts the scores conditioned on both the original reasoning trace and evaluation rationales generated by itself (Ankner et al., 2024; Zhang et al., 2024c).

6 Further improving evaluators

This section discusses some of the recent empirical findings on improving evaluators beyond choosing different types, *e.g.*, training data, input format, and scaling compute.

Validity and utility are complementary Validity measures if the step is logically correct, while utility measures if the step makes progress towards the correct answer. Initially, utility-based process reward models were proposed as an *alternative* for validity, since constructing validity data often requires a costly annotation process (Wang et al., 2024c). Under the hood, there lies an implicit assumption that useful steps are mostly valid.

However, recent works show that the two criteria are rather *complementary*, from training sequence classifiers to using critic models. Zhang et al. (2025) trains a sequence classifier by only considering steps that are both valid (judged by critic models) and useful (by MCTS-based rollouts) steps as positive, substantially improving performance over baselines that only consider validity or utility (Figure 4, *Sequence Classifiers*). Sun et al. (2024); Kim et al. (2025b) has also shown that averaging validity and utility scores from critic models

[2] While *reward model* generally refers to any model that predicts the desirability of an action in reinforcement learning, the term '(process/outcome) reward model' in the context of reasoning trace evaluation often refers to the sequence classifier architecture.

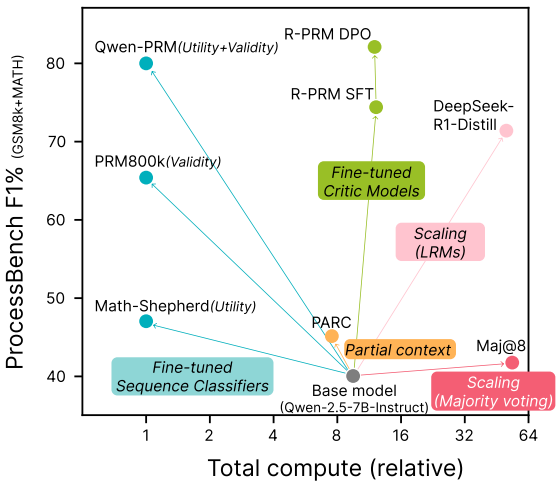

Figure 4: Plot of different evaluators introduced in Section 6, plotted by ProcessBench performance (Zheng et al., 2024a) (GSM8k, MATH subsets averaged) versus total compute for evaluating a trace. While these evaluators share the same base model (Qwen-2.5-7B), they improve the base model's trace evaluation capability in different ways. Details can be found in Appendix D.

results in better performance than individual scores in Best-of-N decoding.

The misalignment between validity and utility is mainly caused by steps that are logically wrong but reach the correct answer (Zheng et al., 2024a; Wang et al., 2025b; Kim et al., 2025b). These invalid but useful steps, also known as *unfaithful reasoning* of LLMs (Lyu et al., 2023; Schnitzler et al., 2024), might lead to overestimation of reasoning ability (Lyu et al., 2023; Petrov et al., 2025).

Partial context allows efficient and accurate evaluation Validity and coherence evaluate a step based on its previous steps. The most intuitive way is to use the *full context* (all preceding steps). However, this approach is not feasible when the trace exceeds the context length of the evaluators, *e.g.*, large reasoning models' traces are often too long to apply critic models (Kim et al., 2025b).

An alternative solution is to use a *partial context*, where only relevant parts of the query and preceding steps are selected and passed to the evaluator (Ling et al., 2023; Mukherjee et al., 2025). These works first construct a directed *entailment graph*, and evaluate the step only based on the identified premises. This allows evaluators to use shorter context, which is both computationally efficient (Ling et al., 2023) and even more accurate as distractors are removed from the context (Mukherjee et al., 2025) (Figure 4, *Partial context*). Moreover, the graph structure also distinguishes direct errors (premises are valid but the reasoning is invalid) and accumulated errors (premises are invalid but reasoning is valid) (Mukherjee et al., 2025).

Test-time scaling improves evaluator performance Test-time scaling is a general paradigm where investing more test-time compute leads to improved performance. Test-time compute can be scaled in diverse directions, such as sampling the output multiple times (Wang et al., 2023b; Yao et al., 2023) or generating more tokens during a single inference (Snell et al., 2024; Qwen-Team, 2024; DeepSeek-AI, 2025).

This paradigm can be extended to critic models that reason on reasoning traces, especially in meta-evaluation benchmarks. When applying majority voting of independently sampled K evaluator scores in generative models (critic model, generative verifiers), the accuracy in predicting incorrect steps increases linearly with the scale of $\log K$ (Singhi et al., 2025; Kim et al., 2025b; She et al., 2025) (Figure 4, *Majority voting*). Furthermore, using large reasoning models (LRMs) with stronger reasoning capability by generating longer traces (Zheng et al., 2024a; Kim et al., 2025b; Khalifa et al., 2025) leads to significant improvement in error detection (Figure 4, *Scaling (LRMs)*).

In verifier-guided search settings, one can either scale exploration or scale evaluation. For instance, in Best-of-N decoding, one can increase the number of responses or use critic models that produce longer outputs. What is the optimal strategy with a constrained computing budget? For relatively weaker evaluators, simple majority voting (Wang et al., 2023b) often outperforms verifier-guided search (Zhang et al., 2025; Singhi et al., 2025). However, using stronger evaluators, *e.g.*, sequence classifiers with better training data (Zhang et al., 2025) or critic models with stronger reasoning capabilities (Khalifa et al., 2025; Kim et al., 2025b) for Best-of-N can effectively outperform majority voting using the same computation budget.

7 Future directions

Evaluating real-world reasoning traces with external knowledge Existing datasets for reasoning trace evaluation are mostly restricted to simple factual reasoning (*e.g.* factual multi-hop question answering) or self-contained problems (*e.g.* math problems). However, many realistic reasoning tasks such as repository-level coding (Zhang

et al., 2023a), medicine (Savage et al., 2024), and law (Holzenberger and Van Durme, 2021; Kim et al., 2024b) require external up-to-date knowledge retrieval-augmented generation (Lewis et al., 2020). Developing evaluators and meta-evaluation benchmarks for these tasks will significantly enhance the applicability of reasoning trace evaluation in more realistic scenarios.

Evaluating long, complex reasoning traces Following OpenAI o1 (OpenAI, 2024b), numerous large reasoning models (LRMs) that generate long, complex traces involving self-verification and backtracking were introduced (DeepSeek-AI, 2025; Muennighoff et al., 2025; Gandhi et al., 2025). However, existing evaluators are not suitable for these complex traces. For instance, assigning a single scalar score (*e.g.*, sequence classifiers) will make invalid steps corrected afterwards (*Wait, this reasoning is not correct.*) and ones not corrected indistinguishable. Since LRM reasoning traces can contain critical errors (Petrov et al., 2025; Chen et al., 2025b), the effort to develop evaluation resources for such traces will lead to a better understanding of LRMs' behaviors and further improvement in their performance and credibility.

Advanced methods for finding premises. NLI-based validity and coherence evaluation significantly benefit from determining the previous steps that the current step uses as a premise (Mukherjee et al., 2025). However, finding such steps is not a trivial task. ROSCOE (Golovneva et al., 2023a) uses the minimum NLI score of all (previous step, current step) combinations, which ignores cases where a step has multiple premises. Recent works (Ling et al., 2023; Tyen et al., 2024; Mukherjee et al., 2025) make the reasoner LLM annotate the premises of the given step. Plausible but underexplored approaches include applying *uncertainty-based methods* (Chen et al., 2023; Wu et al., 2024a) or training a parser that annotates the logical dependencies between steps as graphs (Lee et al., 2025b).

Symbol-grounded evaluation of reasoning traces Reasoning tasks often have a symbolic ground truth solution. For instance, deductive reasoning tasks can be represented with formal logic, and arithmetic problems can be expressed as a series of equations or symbolic theorems. These solutions provide precise, formal ways to define evaluators, including validity and utility (progress). However, not much work has been done to exploit the parallel between reasoning traces and the underlying symbolic solution. While several rule-based approaches parse reasoning traces for evaluation in relatively simpler reasoning tasks (Saparov and He, 2023; Nguyen et al., 2024; Li et al., 2023b), no attempts have been made to extend this paradigm to evaluate reasoning traces for more complex and realistic tasks like first-order logic reasoning (Han et al., 2024a,b) and formal math reasoning that use interactive theorem provers (*e.g.*, Lean, Isabelle) (Yang et al., 2023; Gao et al., 2024c).

Rubric-based evaluation for complex and expert-level tasks. Existing evaluators often apply *identical* evaluations for all reasoning trace, *e.g.*, using the same LLM-as-a-judge prompt for all inputs. However, as the reasoning tasks require more domain knowledge and expertise, there is an increasing need for highly specific **rubrics** for evaluating reasoning traces (Kim et al., 2025a). For instance, one can calculate the sum of an arithmetic sequence by adding all terms one by one or finding a general term; the problem-specific rubrics explicitly prefer the latter. However, manual rubric generation is costly and less scalable, which motivates automatic extraction/generation of high-quality reasoning trace rubrics. AutoRace (Hao et al., 2024) aims to generate rubrics automatically based on incorrect responses, while RaR (Gunjal et al., 2025) extracts checklist-style rubrics from ground-truth biomedical documents. Still, automatically obtaining expert-level, high-quality rubrics for more diverse reasoning tasks remains an open question.

8 Conclusion

This survey aims to organize existing criteria and methods for step-by-step reasoning evaluation, which is crucial for understanding and improving LLM's reasoning capabilities. We provide a unified taxonomy for evaluation criteria, a comprehensive review of existing evaluators and their implementation, and examine recent directions on how to improve these evaluators.

Still, diverse challenges remain in evaluating step-by-step reasoning traces. As new reasoning tasks and methods emerge, existing evaluators often become obsolete for evaluating complex reasoning traces from new tasks and models. As LLMs are now involved in challenging and high-stakes reasoning tasks in the real world, understanding the nature of their errors and precisely evaluating the reasoning trace will remain important.

9 Limitation

References This survey includes an extensive list of recent publications (mostly between 2022 and 2025) on reasoning trace evaluation, sourced from *ACL, EMNLP, NeurIPS, and arXiv preprints, *etc*. While there might be missing references due to the sheer volume of works produced in this field, we will continue to update missing references and newly released impactful works that contribute to the field.

Survey on diverse empirical results While Figure 4 contains a controlled comparison between different approaches like training sequence classifier with different data, using partial context, or applying test-time scaling techniques, the comparison is limited to ProcessBench results for two reasons: (1) While most paper report reasoning performance improvement results (Section 2.3), these results are often not directly comparable because they make use of different base model, which strongly affect the overall performance. (2) Other meta-evaluation benchmarks than ProcessBench (Jacovi et al., 2024; Zeng et al., 2024a; Song et al., 2025) have not been applied to diverse evaluator implementations at the time of writing.

10 Acknowledgements

We thank Sagnik Mukherjee for his valuable help in revising the paper and sharing data for PARC in Section 6, which greatly improved the completeness of this work.

References

Shayan Ali Akbar, Md Mosharaf Hossain, Tess Wood, Si-Chi Chin, Erica M Salinas, Victor Alvarez, and Erwin Cornejo. 2024. HalluMeasure: Fine-grained hallucination measurement using chain-of-thought reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15020–15037, Miami, Florida, USA. Association for Computational Linguistics.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. 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 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report. *Preprint*, arXiv:2305.10403.

Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D. Chang, and Prithviraj Ammanabrolu. 2024. Critique-out-loud reward models. *Preprint*, arXiv:2408.11791.

Sourav Banerjee, Ayushi Agarwal, and Saloni Singla. 2024. Llms will always hallucinate, and we need to live with this. *Preprint*, arXiv:2409.05746.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *Preprint*, arXiv:1911.11641.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Madry. 2024. Mle-bench: Evaluating machine learning agents on machine learning engineering. *CoRR*, abs/2410.07095.

Hanjie Chen, Faeze Brahman, Xiang Ren, Yangfeng Ji, Yejin Choi, and Swabha Swayamdipta. 2023. REV: Information-theoretic evaluation of free-text rationales. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2007–2030, Toronto, Canada. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, Hanghang Tong, and Heng Ji. 2025a. Rm-r1: Reward modeling as reasoning. *Preprint*, arXiv:2505.02387.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Sam Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. 2025b. Reasoning models don't always say what they think.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2022. Finqa: A dataset of numerical reasoning over financial data. *Preprint*, arXiv:2109.00122.

Cheng-Han Chiang and Hung-yi Lee. 2024. Over-reasoning 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.

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.

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *Preprint*, arXiv:2208.14271.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. 2025. Process reinforcement through implicit rewards. *Preprint*, arXiv:2502.01456.

Ning Dai, Zheng Wu, Renjie Zheng, Ziyun Wei, Wenlei Shi, Xing Jin, Guanlin Liu, Chen Dun, Liang Huang, and Lin Yan. 2025. Process supervision-guided policy optimization for code generation. *Preprint*, arXiv:2410.17621.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7358–7370, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lizhou Fan, Wenyue Hua, Lingyao Li, Haoyang Ling, and Yongfeng Zhang. 2024. Nphardeval: Dynamic benchmark on reasoning ability of large language models via complexity classes. *Preprint*, arXiv:2312.14890.

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

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *Preprint*, arXiv:2503.01307.

Bofei Gao, Zefan Cai, Runxin Xu, Peiyi Wang, Ce Zheng, Runji Lin, Keming Lu, Dayiheng Liu, Chang Zhou, Wen Xiao, Junjie Hu, Tianyu Liu, and Baobao Chang. 2024a. Llm critics help catch bugs in mathematics: Towards a better mathematical verifier with natural language feedback. *Preprint*, arXiv:2406.14024.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2024b. Omni-math: A universal olympiad level mathematic benchmark for large language models. *Preprint*, arXiv:2410.07985.

Guoxiong Gao, Yutong Wang, Jiedong Jiang, Qi Gao, Zihan Qin, Tianyi Xu, and Bin Dong. 2024c. Herald: A natural language annotated lean 4 dataset. *Preprint*, arXiv:2410.10878.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024d. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

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.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreepranav Varma Enugandla, and Mark Wildon. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *Preprint*, arXiv:2411.04872.

Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023a. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Olga Golovneva, Sean O'Brien, Ramakanth Pasunuru, Tianlu Wang, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023b. Pathfinder: Guided search over multi-step reasoning paths. *Preprint*, arXiv:2312.05180.

Xinyan Guan, Yanjiang Liu, Xinyu Lu, Boxi Cao, Ben He, Xianpei Han, Le Sun, Jie Lou, Bowen Yu, Yaojie Lu, and Hongyu Lin. 2024. Search, verify and feedback: Towards next generation post-training paradigm of foundation models via verifier engineering. *Preprint*, arXiv:2411.11504.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Preprint*, arXiv:2308.11462.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. 2025. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *Preprint*, arXiv:2507.17746.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander Fabbri, Wojciech Maciej Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. 2024a. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.

Simeng Han, Aaron Yu, Rui Shen, Zhenting Qi, Martin Riddell, Wenfei Zhou, Yujie Qiao, Yilun Zhao, Semih Yavuz, Ye Liu, Shafiq Joty, Yingbo Zhou,

Caiming Xiong, Dragomir Radev, Rex Ying, and Arman Cohan. 2024b. P-FOLIO: Evaluating and improving logical reasoning with abundant human-written reasoning chains. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16553–16565, Miami, Florida, USA. Association for Computational Linguistics.

Shibo Hao, Yi Gu, Haotian Luo, Tianyang Liu, Xiyan Shao, Xinyuan Wang, Shuhua Xie, Haodi Ma, Adithya Samavedhi, Qiyue Gao, Zhen Wang, and Zhiting Hu. 2024. Llm reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models. *Preprint*, arXiv:2404.05221.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024a. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand. Association for Computational Linguistics.

Mingqian He, Yongliang Shen, Wenqi Zhang, Zeqi Tan, and Weiming Lu. 2024b. Advancing process verification for large language models via tree-based preference learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2086–2099, Miami, Florida, USA. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher Manning. 2021. Conditional probing: measuring usable information beyond a baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1626–1639, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2020. A dataset for statutory reasoning in tax law entailment and question answering. *Preprint*, arXiv:2005.05257.

Nils Holzenberger and Benjamin Van Durme. 2021. Factoring statutory reasoning as language understanding challenges. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2742–2758, Online. Association for Computational Linguistics.

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.

Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei W Koh, and Bryan Hooi. 2024. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in llms. *Advances in Neural Information Processing Systems*, 37:24181–24215.

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

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2024b. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

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.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. Swe-bench: Can language models resolve real-world github issues? *Preprint*, arXiv:2310.06770.

Liwei Kang, Zirui Zhao, David Hsu, and Wee Sun Lee. 2024. On the empirical complexity of reasoning and planning in llms. *Preprint*, arXiv:2404.11041.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. 2025. Process reward models that think. *Preprint*, arXiv:2504.16828.

Seungone Kim, Jamin Shin, Yejin Choi, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024a. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, Sue Hyun Park, Hyeonbin Hwang, Jinkyung Jo, Hyowon Cho, Haebin Shin, Seongyun Lee, Hanseok Oh, Noah Lee,

Namgyu Ho, Se June Joo, Miyoung Ko, Yoonjoo Lee, Hyungjoo Chae, Jamin Shin, Joel Jang, Seonghyeon Ye, Bill Yuchen Lin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2025a. The BiGGen bench: A principled benchmark for fine-grained evaluation of language models with language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5877–5919, Albuquerque, New Mexico. Association for Computational Linguistics.

Seungone Kim, Ian Wu, Jinu Lee, Xiang Yue, Seongyun Lee, Mingyeong Moon, Kiril Gashteovski, Carolin Lawrence, Julia Hockenmaier, Graham Neubig, and Sean Welleck. 2025b. Scaling evaluation-time compute with reasoning models as process evaluators. *Preprint*, arXiv:2503.19877.

Yeeun Kim, Youngrok Choi, Eunkyung Choi, JinHwan Choi, Hai Jin Park, and Wonseok Hwang. 2024b. Developing a pragmatic benchmark for assessing Korean legal language understanding in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5573–5595, Miami, Florida, USA. 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.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California. Association for Computational Linguistics.

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.

Eldar Kurtic, Amir Moeini, and Dan Alistarh. 2024. Mathador-LM: A dynamic benchmark for mathematical reasoning on large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17020–17027, Miami, Florida, USA. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. 2024. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *Preprint*, arXiv:2406.18629.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, Lester James Validad Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. RewardBench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797, Albuquerque, New Mexico. Association for Computational Linguistics.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. 2023. Measuring faithfulness in chain-of-thought reasoning. *Preprint*, arXiv:2307.13702.

Jinu Lee and Wonseok Hwang. 2025. Symba: Symbolic backward chaining for structured natural language reasoning. *Preprint*, arXiv:2402.12806.

Jinu Lee, Qi Liu, Runzhi Ma, Vincent Han, Ziqi Wang, Heng Ji, and Julia Hockenmaier. 2025a. Entailment-preserving first-order logic representations in natural language entailment. *Preprint*, arXiv:2502.16757.

Jinu Lee, Sagnik Mukherjee, Dilek Hakkani-Tur, and Julia Hockenmaier. 2025b. Reasoningflow: Semantic structure of complex reasoning traces. *Preprint*, arXiv:2506.02532.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. Halueval: A large-scale hallucination evaluation benchmark for large language models. *Preprint*, arXiv:2305.11747.

Ruosen Li, Zimu Wang, Son Tran, Lei Xia, and Xinya Du. 2024a. Meqa: A benchmark for multi-hop event-centric question answering with explanations. In *Advances in Neural Information Processing Systems*, volume 37, pages 126835–126862. Curran Associates, Inc.

Xiang Li, Zhenyu Li, Chen Shi, Yong Xu, Qing Du, Mingkui Tan, and Jun Huang. 2024b. AlphaFin: Benchmarking financial analysis with

retrieval-augmented stock-chain framework. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 773–783, Torino, Italia. ELRA and ICCL.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023b. 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.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. 2022. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Zicheng Lin, Zhibin Gou, Tian Liang, Ruilin Luo, Haowei Liu, and Yujiu Yang. 2024. Criticbench: Benchmarking llms for critique-correct reasoning. *Preprint*, arXiv:2402.14809.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. In *Advances in Neural Information Processing Systems*, volume 36, pages 36407–36433. Curran Associates, Inc.

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.

Dakuan Lu, Xiaoyu Tan, Rui Xu, Tianchu Yao, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi. 2025. Scp-116k: A high-quality problem-solution dataset and a generalized pipeline for automated extraction in the higher education science domain. *Preprint*, arXiv:2501.15587.

Zimu Lu, Aojun Zhou, Ke Wang, Houxing Ren, Weikang Shi, Junting Pan, Mingjie Zhan, and Hongsheng Li. 2024. Step-controlled dpo: Leveraging stepwise error for enhanced mathematical reasoning. *Preprint*, arXiv:2407.00782.

Junliang Luo, Tianyu Li, Di Wu, Michael Jenkin, Steve Liu, and Gregory Dudek. 2024a. Hallucination detection and hallucination mitigation: An investigation. *arXiv preprint arXiv:2401.08358*.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. 2024b. Improve mathematical reasoning in language models by automated process supervision. *Preprint*, arXiv:2406.06592.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329, Nusa Dua, Bali. Association for Computational Linguistics.

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

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *Preprint*, arXiv:2005.00661.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online. 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. *Preprint*, arXiv:1809.02789.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *Preprint*, arXiv:2410.05229.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

Sagnik Mukherjee, Abhinav Chinta, Takyoung Kim, Tarun Anoop Sharma, and Dilek Hakkani-Tür. 2025. Premise-augmented reasoning chains improve error identification in math reasoning with llms. *Preprint*, arXiv:2502.02362.

Thi 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.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. *Preprint*, arXiv:2401.00396.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176, Singapore. Association for Computational Linguistics.

OpenAI. 2024a. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI. 2024b. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Simon Ott, Konstantin Hebenstreit, Valentin Liévin, Christoffer Egeberg Hother, Milad Moradi, Maximilian Mayrhauser, Robert Praas, Ole Winther, and Matthias Samwald. 2023. Thoughtsource: A central hub for large language model reasoning data. *Scientific Data*, 10(1).

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023a. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

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

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. *Preprint*, arXiv:2404.19733.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. 2024. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15012–15032, Miami, Florida, USA. Association for Computational Linguistics.

Ivo Petrov, Jasper Dekoninck, Lyuben Baltadzhiev, Maria Drencheva, Kristian Minchev, Mislav Balunović, Nikola Jovanović, and Martin Vechev. 2025. Proof or bluff? evaluating llms on 2025 usa math olympiad. *Preprint*, arXiv:2503.21934.

Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. 2023. ReCEval: Evaluating reasoning chains via correctness and informativeness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10066–10086, Singapore. Association for Computational Linguistics.

Zexuan Qiu, Zijing Ou, Bin Wu, Jingjing Li, Aiwei Liu, and Irwin King. 2024. Entropy-based decoding for retrieval-augmented large language models. *Preprint*, arXiv:2406.17519.

Qwen-Team. 2024. QwQ: Reflect Deeply on the Boundaries of the Unknown — qwenlm.github.io. `https://qwenlm.github.io/blog/qwq-32b-preview/`. [Accessed 13-02-2025].

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

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*.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations*.

Thomas Savage, Ashwin Nayak, Robert Gallo, Ekanath Rangan, and Jonathan H Chen. 2024. Diagnostic reasoning prompts reveal the potential for large language model interpretability in medicine. *NPJ Digital Medicine*, 7(1):20.

Julian Schnitzler, Xanh Ho, Jiahao Huang, Florian Boudin, Saku Sugawara, and Akiko Aizawa. 2024. Morehopqa: More than multi-hop reasoning. *Preprint*, arXiv:2406.13397.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2024. Rewarding progress: Scaling automated process verifiers for llm reasoning. *Preprint*, arXiv:2410.08146.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Shuaijie She, Junxiao Liu, Yifeng Liu, Jiajun Chen, Xin Huang, and Shujian Huang. 2025. R-prm: Reasoning-driven process reward modeling. *Preprint*, arXiv:2503.21295.

Nishad Singhi, Hritik Bansal, Arian Hosseini, Aditya Grover, Kai-Wei Chang, Marcus Rohrbach, and Anna Rohrbach. 2025. When to solve, when to verify: Compute-optimal problem solving and generative verification for llm reasoning. *Preprint*, arXiv:2504.01005.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4515, Hong Kong, China. Association for Computational Linguistics.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. 2025. Prmbench: A fine-grained and challenging benchmark for process-level reward models. *Preprint*, arXiv:2501.03124.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *Preprint*, arXiv:2409.12183.

Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. 2024. Easy-to-hard generalization: Scalable alignment beyond human supervision. *Preprint*, arXiv:2403.09472.

Simon Šuster and Walter Daelemans. 2018. CliCR: a dataset of clinical case reports for machine reading comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1551–1563, New Orleans, Louisiana. Association for Computational Linguistics.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *Preprint*, arXiv:2210.09261.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

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.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. 2021. Diagnosing the first-order logical reasoning ability through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3738–3747, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Armin Toroghi, Willis Guo, Mohammad Mahdi Abdollah Pour, and Scott Sanner. 2024. Right for right reasons: Large language models for verifiable commonsense knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6601–6633, Miami, Florida, USA. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Nemika Tyagi, Mihir Parmar, Mohith Kulkarni, Aswin Rrv, Nisarg Patel, Mutsumi Nakamura, Arindam Mitra, and Chitta Baral. 2024. Step-by-step reasoning to solve grid puzzles: Where do LLMs falter? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19898–19915, Miami, Florida, USA. 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.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, Songyang Gao, Nuo Xu, Yuhao Zhou, Xiaoran Fan, Zhiheng Xi, Jun Zhao, Xiao Wang, Tao Ji, Hang Yan, Lixing Shen, Zhan Chen, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. 2024a. Secrets of rlhf in large language models part ii: Reward modeling. *Preprint*, arXiv:2401.06080.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023a. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Hanyin Wang, Chufan Gao, Qiping Xu, Bolun Liu, Guleid Hussein, Hariprasad Korsapati, Mohamad El Labban, Kingsley Iheasirim, Mohamed Hassan, Gokhan Anil, Brian Bartlett, and Jimeng Sun. 2025a. Process-supervised reward models for verifying clinical note generation: A scalable approach guided by domain expertise. *Preprint*, arXiv:2412.12583.

Jianing Wang, Qiushi Sun, Xiang Li, and Ming Gao. 2024b. Boosting language models reasoning with chain-of-knowledge prompting. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4958–4981, Bangkok, Thailand. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024c. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yu Wang, Nan Yang, Liang Wang, and Furu Wei. 2025b. Examining false positives under inference scaling for mathematical reasoning. *Preprint*, arXiv:2502.06217.

Zecheng Wang, Chunshan Li, Zhao Yang, Qingbin Liu, Yanchao Hao, Xi Chen, Dianhui Chu, and Dianbo Sui. 2024d. Analyzing chain-of-thought prompting in black-box large language models via estimated V-information. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 893–903, Torino, Italia. ELRA and ICCL.

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.

Ting-Ruen Wei, Haowei Liu, Xuyang Wu, and Yi Fang. 2025. A survey on feedback-based multi-step reasoning for large language models on mathematics. *Preprint*, arXiv:2502.14333.

Di Wu, Jia-Chen Gu, Fan Yin, Nanyun Peng, and Kai-Wei Chang. 2024a. Synchronous faithfulness monitoring for trustworthy retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9390–9406, Miami, Florida, USA. Association for Computational Linguistics.

Jian Wu, Linyi Yang, Zhen Wang, Manabu Okumura, and Yue Zhang. 2024b. Cofca: A step-wise counterfactual multi-hop qa benchmark. *Preprint*, arXiv:2402.11924.

Junda Wu, Xintong Li, Ruoyu Wang, Yu Xia, Yuxin Xiong, Jianing Wang, Tong Yu, Xiang Chen, Branislav Kveton, Lina Yao, Jingbo Shang, and Julian McAuley. 2024c. Ocean: Offline chain-of-thought evaluation and alignment in large language models. *Preprint*, arXiv:2410.23703.

Yexin Wu, Zhuosheng Zhang, and Hai Zhao. 2024d. Mitigating misleading chain-of-thought reasoning with selective filtering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11325–11340, Torino, Italia. ELRA and ICCL.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2025. Evaluating mathematical reasoning beyond accuracy. *Preprint*, arXiv:2404.05692.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024. Monte carlo tree search boosts reasoning via iterative preference learning. *Preprint*, arXiv:2405.00451.

Zhengnan Xie, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter Jansen. 2020. WorldTree v2: A corpus of science-domain structured explanations and inference patterns supporting multi-hop inference. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5456–5473, Marseille, France. European Language Resources Association.

Kaiyu Yang, Aidan Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. 2023. LeanDojo: Theorem proving with retrieval-augmented language models. In *Neural Information Processing Systems (NeurIPS)*.

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, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. 2024. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*.

Thomas Zeng, Shuibai Zhang, Shutong Wu, Christian Classen, Daewon Chae, Ethan Ewer, Minjae Lee, Heeju Kim, Wonjun Kang, Jackson Kunde, Ying Fan, Jungtaek Kim, Hyung Il Koo, Kannan Ramchandran, Dimitris Papailiopoulos, and Kangwook Lee. 2025. Versaprm: Multi-domain process reward model via synthetic reasoning data. *Preprint*, arXiv:2502.06737.

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. In *Advances in Neural Information Processing Systems*, volume 37, pages 119466–119546. Curran Associates, Inc.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. Rest-mcts* : Llm self-training via process reward guided tree search. In *Advances in Neural Information Processing Systems*, volume 37, pages 64735–64772. Curran Associates, Inc.

Fengji Zhang, Bei Chen, Yue Zhang, Jacky Keung, Jin Liu, Daoguang Zan, Yi Mao, Jian-Guang Lou, and Weizhu Chen. 2023a. RepoCoder: Repository-level code completion through iterative retrieval and generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2471–2484, Singapore. Association for Computational Linguistics.

Jiaxin Zhang, Zhong-Zhi Li, Ming-Liang Zhang, Fei Yin, Cheng-Lin Liu, and Yashar Moshfeghi. 2024b. GeoEval: Benchmark for evaluating LLMs and multi-modal models on geometry problem-solving. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1258–1276, Bangkok, Thailand. Association for Computational Linguistics.

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

Tianhang Zhang, Lin Qiu, Qipeng Guo, Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou, Xinbing Wang, and Luoyi Fu. 2023b. Enhancing uncertainty-based hallucination detection with stronger focus. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 915–932, Singapore. Association for Computational Linguistics.

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. 2024d. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *Preprint*, arXiv:2406.09136.

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. 2024a. 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 Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Xin Zheng, Jie Lou, Boxi Cao, Xueru Wen, Yuqiu Ji, Hongyu Lin, Yaojie Lu, Xianpei Han, Debing Zhang, and Le Sun. 2024b. Critic-cot: Boosting the reasoning abilities of large language model via chain-of-thoughts critic. *Preprint*, arXiv:2408.16326.

Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. 2021. Ar-lsat: Investigating analytical reasoning of text. *Preprint*, arXiv:2104.06598.

Zhanke Zhou, Rong Tao, Jianing Zhu, Yiwen Luo, Zengmao Wang, and Bo Han. 2024. Can language models perform robust reasoning in chain-of-thought prompting with noisy rationales? *Preprint*, arXiv:2410.23856.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024a. FanOutQA: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 18–37, Bangkok, Thailand. Association for Computational Linguistics.

Tinghui Zhu, Kai Zhang, Jian Xie, and Yu Su. 2024b. Deductive beam search: Decoding deducible rationale for chain-of-thought reasoning. *Preprint*, arXiv:2401.17686.

Yuqi Zhu, Ge Li, Xue Jiang, Jia Li, Hong Mei, Zhi Jin, and Yihong Dong. 2025. Uncertainty-guided chain-of-thought for code generation with llms. *Preprint*, arXiv:2503.15341.

Yuxin Zuo, Shang Qu, Yifei Li, Zhangren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. 2025. Medxpertqa: Benchmarking expert-level medical reasoning and understanding. *Preprint*, arXiv:2501.18362.

A Tasks

This section describes different reasoning tasks and datasets in more detail. While all reasoning tasks fundamentally share the same criteria, literature about a specific task has focused on one criterion over others. For instance, evaluators for factual reasoning tasks often emphasized detecting infactual statements, while evaluators for math reasoning tasks aimed for invalid statements. These discrepancies are one of the fundamental causes of the divergence of the terminologies and definitions in the field.

A.1 Multi-hop Question Answering

Multi-hop question answering (MHQA) tasks require taking information from multiple sources to derive the correct answer (Yang et al., 2018). MHQA is often divided into two subcategories, **factual reasoning** and **commonsense reasoning**.

Answering factual MHQAs can be seen as finding the sequence of *bridging entities* that leads to the final answer (Yang et al., 2018; Talmor and Berant, 2018; Kwiatkowski et al., 2019). For example, to solve a factual MHQA question *"The Argentine PGA Championship record holder has won how many tournaments worldwide?"*, one must first find who the Argentine PGA championship record holder is (bridging entity) and determine how many tournaments he has won worldwide. As bridging entity identification does not require sophisticated reasoning ability compared to other tasks, reasoning trace evaluation on factual MHQA mostly focuses on the factuality based on semantic alignment between the query (retrieved documents) and the trace (Golovneva et al., 2023a).

In contrast, an inference step in commonsense MHQAs (Clark et al., 2018; Mihaylov et al., 2018; Talmor et al., 2019; Bisk et al., 2019; Geva et al., 2021; Trivedi et al., 2022) can require information that is not present in the query. The form of such commonsense knowledge can be diverse, ranging from well-known facts (*Paris is in France.*) to logical rules (*If A was born after B was dead, they have never met each other*). Due to these implicit steps, factuality, validity, and coherence are often hard to separate in evaluating commonsense reasoning traces (Jacovi et al., 2024; Zeng et al., 2024b). Furthermore, due to the inherent subjectiveness of validity and coherence in commonsense reasoning, there might be non-negligible inter-annotator disagreement on certain questions (Jacovi et al., 2024)

LLMs are known to achieve strong performance in challenging MHQA datasets such as ARC-Challenge and PIQA, sometimes exceeding human performance (OpenAI, 2024a; Anil et al., 2023). However, multiple studies report that even modern LLMs like GPT-4 (OpenAI, 2024a) are vulnerable to errors, such as failing to correctly adhere to long evidence (Zhu et al., 2024a), leveraging shortcuts (Schnitzler et al., 2024), or ignoring the temporal relation between events (Li et al., 2024a). Therefore, identifying and categorizing mistakes made by LLMs in these tasks is still an important goal.

A.2 Symbolic Reasoning

Since the discovery of Chain-of-thought prompting (Wei et al., 2022; Kojima et al., 2022), step-by-step reasoning largely expanded LLMs' ability to solve symbolic reasoning tasks[3] such as **mathematical reasoning**, **logical reasoning**, and **algorithmic reasoning**. As the final answer and the reasoning process are highly objective in these tasks, utility and validity are the two most popular criteria for evaluating reasoning traces from symbolic tasks.

Arithmetic reasoning, where the model has to predict the correct answer from arithmetic word problems, is the most renowned variant of math reasoning. Popular benchmarks include MathQA (Amini et al., 2019) and GSM8k (Cobbe et al., 2021), which provide long, diverse natural language queries in contrast to relatively synthetic, simple benchmarks (Koncel-Kedziorski et al., 2016; Miao et al., 2020). Game of 24 (Yao et al., 2023) and Mathador (Kurtic et al., 2024) ask to combine given numbers and arithmetic operations to generate the target number, requiring exploration and backtracking in the exponential solution space.

The recent saturation of LLMs in arithmetic word problems facilitated more challenging **mathematical reasoning** benchmarks from math competitions and university textbooks, covering fields like calculus, probability, statistics, geometry, number theory, and more (He et al., 2024a; Gao et al., 2024b; Glazer et al., 2024; Zhang et al., 2024b). While these benchmarks were highly challenging to the state-of-the-art LLMs of the time of release, recently emerging large reasoning models (OpenAI, 2024b; Qwen-Team, 2024; DeepSeek-AI, 2025) achieve unprecedented performance in these benchmarks by generating long reasoning traces often with self-verification and backtracking.

Deductive logical reasoning (Tafjord et al., 2021; Tian et al., 2021; Saparov and He, 2023; Han et al., 2024a) mainly focuses on logical deduction, where one should repeatedly apply the general rules to specific facts as in classical syllogism. **Constraint-based reasoning** (Zhong et al., 2021; Tyagi et al., 2024) is a variant of deductive reasoning where one must find the solution that satisfies the provided initial constraints (*e.g.*, grid puzzles (Zhong et al., 2021)). As these datasets are easy to solve in a symbolic form like logic programming (Saparov and He, 2023; Pan et al., 2023a; Olausson et al., 2023; Lee and Hwang, 2025) but harder in natural language due to the size of the search space (Kang et al., 2024), they have served as a diagnostic benchmark for understanding and analyzing the complex reasoning capability of large language models (Sinha et al., 2019; Saparov and He, 2023; Han et al., 2024a). However, as these datasets are often synthetically generated from their symbolic representations, they might not fully generalize to real-world problems with linguistic diversity and commonsense.

Finally, **algorithmic reasoning** tasks include manipulating strings and data structures, such as concatenating the last letters of the given words (Wei et al., 2022) or completing the incomplete Dyck language. BIG-Bench-Hard (BBH; Suzgun et al. (2022)) and NPHardEval (Fan et al., 2024) include 11 and 9 algorithmic reasoning tasks, respectively, which are challenging for modern LLMs like GPT-4 and PaLM-540B.

A.3 Others

Science reasoning tasks lie between factual/commonsense reasoning tasks and symbolic reasoning tasks, as they often require understanding complicated facts combined with world knowledge and performing precise math/logical reasoning (Hendrycks et al., 2021; Rein et al., 2024; He et al., 2024a; Lu et al., 2025). The most popular benchmark in this field, GPQA-Diamond (Rein et al., 2024), contains 546 questions from physics, chemistry, and biology, where human experts only get 65% of the problems correct.

Expert-domain reasoning includes domain-specific reasoning tasks that often require significant expertise in the field, *e.g.*, biomedical reason-

[3]While symbolic reasoning may strictly refer to *algorithmic reasoning* in some literature (Wei et al., 2022; Suzgun et al., 2022), we adopt the broader sense including math and logical reasoning that can be readily expressed in symbols (*e.g.*, equation, logic) (Sprague et al., 2024).

ing (Šuster and Daelemans, 2018; Savage et al., 2024; Zuo et al., 2025), legal reasoning (Holzenberger et al., 2020; Guha et al., 2023; Kim et al., 2024b), and financial reasoning (Chen et al., 2022; Li et al., 2024b). These tasks require both domain-specific knowledge and reasoning strategies, posing a significant challenge to modern language models (Zuo et al., 2025; Li et al., 2024b). However, due to the high cost of expert annotation, existing methods often oversimplify real-world challenges (Holzenberger and Van Durme, 2021; Guha et al., 2023); consequently, the demand for resources that closely reflect real-world expert applications is rising.

Programming/coding is closely related to algorithmic reasoning. Popular benchmarks regarding programming include *competitive coding*, where one has to solve an algorithm problem given in natural language and test codes (Chen et al., 2021; Li et al., 2022), and *practical coding* that covers tasks of software engineers and developers (Zhang et al., 2023a; Jimenez et al., 2024; Chan et al., 2024). Programming differs from other reasoning tasks in various aspects: (1) codes are strictly constrained by predefined syntax and semantics, and (2) the result is evaluated by the execution result rather than the code itself. These constraints make (1) segmenting the trace (code) into steps and (2) applying metrics that require explicitly stated answers, *i.e.*, $\mathcal{V}$-information, difficult than in natural language reasoning traces. Therefore, most evaluators specialized in code focus on trace-level utility rather than step-wise evaluation, defined as the pass rate of predefined unit tests (Dai et al., 2025).

B Appendix for Meta-evaluation Datasets

This appendix includes discussions on the dataset construction process, with a focus on data annotation and label types. A summary of existing datasets can be found in Table 4.

B.1 Data collection process

B.1.1 Labeling methods

Human annotation The most straightforward approach to decide the ground truth label is to use *human evalua tion* (Lightman et al., 2024; Jacovi et al., 2024; Zeng et al., 2024a; Zheng et al., 2024a). The largest human annotation experiment was conducted by Lightman et al. (2024), where crowdsourced annotators labeled the validity of 800k steps (75k reasoning traces). Due to the sheer volume of annotation, an active learning strategy was used; the annotators were requested to annotate *hard* samples (the final answer is incorrect but judged as valid by the reward model), which were added to the training data for the next version of the reward model.

LLM annotation As a cheap alternative for human evaluation, LLM-as-a-judge is often used to generate labels (Gao et al., 2024a; Zhang et al., 2025). However, LLM-assigned labels are not fully credible, given that state-of-the-art LLMs still make errors in human-annotated datasets (Zheng et al., 2024a; Kim et al., 2025b). Therefore, LLM-annotated data is often used to augment the training data rather than for meta-evaluation purposes.

Perturbation Another method to create positive and negative samples is to insert errors into correct reasoning traces. For instance, Zhu et al. (2024b); Lu et al. (2024) samples traces that reach the correct answer, and prompts an LLM to introduce a predefined form of perturbation to the reasoning trace. This allows easy sampling of diverse erroneous traces that can improve the robustness of evaluators, but using human-defined errors might not correctly reflect the true distribution of LLM-generated errors.

Step-level utility Some datasets use step-level utility as their labels. The most prominent approach is *Monte Carlo Tree Search* (Wang et al., 2024c), where the step-level utility is measured by sampling *rollouts* from a step and checking if they reach the correct answer. However, to increase the efficiency of the search for negative labels (low utility), Luo et al. (2024b); Dai et al. (2025) implements a binary search algorithm to locate the first step with low utility. One notable variant of step-level utility labels is *advantage*, where the evaluators are not trained to predict the expected reward of each node but the *change* in the expected rewards before and after generating the step (Setlur et al., 2024).

Trace-level utility The coarsest label is the trace-level utility, simply measured by the correctness of the final answer (Lambert et al., 2025).

Both trace-level and step-level utilities do not require human annotation other than the final answer, which is much cheaper to obtain than human annotations (Wang et al., 2024c). However, they cannot serve as a reliable proxy of factuality/coherence/validity due to *unfaithful reasoning*, where traces that reach the correct answer (high

Dataset	Train	Eval	Domain	Criteria	# Trace	Human
ROSCOE (Golovneva et al., 2023b)		●	Math, Common	FVU	1.0k	●
RAGTruth† (Niu et al., 2024)	●	●	Fact	F	5.9k	●
HaluEval† (Li et al., 2023a)	●	●	Fact	F	10k	▲
Math-Shepherd (Wang et al., 2024c)	●		Math	U	440k	×
PRM800k (Lightman et al., 2024)	●	●	Math	V	75k	●
REVEAL (Jacovi et al., 2024)		●	Common	FVC	3.4k	●
MATH-Minos (Gao et al., 2024a)	●		Math	V	440k	×
SCDPO (Lu et al., 2024)	●		Math	U	30k	×
MR-GSM8k (Zeng et al., 2024a)		●	Math	V	3.0k	●
BIG-Bench-Mistake (Tyen et al., 2024)		●	Symbolic	VCU	2.2k	●
CriticBench (Lin et al., 2024)		●	Math, Common, Symbolic	VU	3.8k	×
ProcessBench (Zheng et al., 2024a)		●	Math	V	3.4k	●
MR-Ben (Zeng et al., 2024b)		●	Science, Deductive, Coding	V	6.0k	●
MR-MATH (Xia et al., 2025)		●	Math	VU	0.1k	●
PRMBench (Song et al., 2025)		●	Math	VCU	6.2k	▲
PRM-Clinic (Wang et al., 2025a)		●	Expert(Clinic)	FVC	9.7k	×
VersaPRM (Zeng et al., 2025)	●		Expert	FV	84.1k	×
BiGGenBench† (Kim et al., 2025a)		●	Math, Logic	Custom	0.1k	×

Table 4: List of evaluator training data and meta-evaluation benchmarks. † symbol indicates that the datasets include other tasks, such as summarization, instruction following, *etc*, where the **# Trace** column only counts the reasoning subset. **Train/Eval** columns denote if the dataset is used for training or meta-evaluation. **Domain** indicates what tasks are used to sample the reasoning trace. **Criteria** column shows the criteria used to annotate the data classified according to Section 3, where FVCU stands for factuality, validity, coherence, and utility, respectively. BiGGenBench (Kim et al., 2025a) applies hand-written, query-specific evaluation criteria (**Custom**). **Human** column indicates human annotation, where ● ▲ × denotes full human annotation, automatic annotation/perturbation with human verification, and full LLM-based annotation, respectively.

utility) often include factual/logical errors (Lanham et al., 2023; Lyu et al., 2023; Zheng et al., 2024a; Kim et al., 2025b).

B.1.2 Inter-annotator agreement

While reasoning trace evaluation is considered more objective than other long-text evaluation tasks (*e.g.*, helpfulness, bias/harmfulness, and language proficiency) (Wang et al., 2024a), a certain amount of inter-annotator disagreement is inevitable. Here, we report the trend in inter-annotator agreement observed in existing human annotation works.

Incorrect solutions for harder problems lead to higher disagreement ProcessBench (Zheng et al., 2024a) consolidates the intuitive hypothesis that inter-annotator disagreement grows when the query is difficult and the trace is incorrect in at least one step. Compared to the easiest case (GSM8k queries, correct trace), where three annotators agree in 95.9% of the cases, the hardest case (OmniMATH, incorrect trace) shows only 47.8% of three-annotator agreement.

Inter-annotator disagreement reflects vagueness in natural language In many cases, the disagreement is significantly affected by the linguistic aspects of the reasoning trace. REVEAL (Jacovi et al., 2024) manually classifies steps that aroused disagreement between annotators into 13 distinct categories. Among these, frequent disagreement types like "World knowledge (some world knowledge might not be taken for granted)" and "Unclear reference (one proper noun can refer to multiple real-world entities)" are typical disagreement types observed in simpler *recognizing textual entailment* (natural language inference) tasks (Camburu et al., 2018; Lee et al., 2025a), showing that these vagueness is present even in *minimal* settings.

On the other hand, synthetic, algorithmic reasoning tasks like BIG-Bench-Hard (Suzgun et al., 2022) are linguistically uniform. Consequently, BIG-Bench-Mistake that annotate errors in this benchmark (Tyen et al., 2024) observes near-perfect inter-annotator agreement (Krippendorf's $\alpha > 0.97$), again demonstrating the strong connection between linguistic variation and inter-annotator agreement.

B.2 Label types

Sequence classification The most common label type is sequence classification, where a quality label is assigned to each step/trace. For example, Wang et al. (2024c) assigns binary labels to steps based on the utility, and Lightman

et al. (2024) assigns ternary validity labels (*correct/incorrect/neutral*) obtained by human annotation. The *neutral* label in Lightman et al. (2024) was introduced to absorb ambiguous cases and minimize inter-annotator disagreement; considering it as positive or negative when training the evaluator does not significantly affect the Best-of-N performance (<1.0p).

One caveat of sequence classification is that it is hard to define the labels *after* the first error (propagated error). It is often unclear whether steps that rely on the first erroneous step should be labeled as incorrect (because they rely on incorrect premises) or correct (because the reasoning is correct if assuming the premises are correct) (Jacovi et al., 2024; Mukherjee et al., 2025). Two different label schemas are used to bypass this ambiguity: annotating the *pairwise preference* and annotating the *index of the first erroneous step*.

Preference (win/lose) Reasoning trace evaluation can be formulated as a preference problem (Lai et al., 2024; Lu et al., 2024; Lambert et al., 2025). In this scenario, data points are defined as pairs of reasoning traces, one as the winner and the other as the loser. The pairs are often constructed by sampling two different continuations from a shared prefix or perturbing a correct trace. These data are often used to train the LLM-as-a-value-function models via preference learning algorithms, *e.g.*, DPO (Rafailov et al., 2023).

Identifying first erroneous index Another method is to label the index of the first erroneous step (Zheng et al., 2024a; Zeng et al., 2024a). In this setting, the reasoning trace is given as a list of steps, and the evaluator must predict the index of the first error. If there is no error, the model should predict -1. This setting effectively bypasses the propagated error problem, but converting these labels to binary classification can lead to better performance in sequence classifiers and critic models (Kim et al., 2025b).

C Comparing criteria definitions

C.1 Comparison between proposed definitions

Factuality↔Validity Factuality focuses on the relationship between a step and provided/external knowledge, while validity focuses on the relationship between two model-generated steps. For instance, Given an incorrect step *Albert Einstein died in 1965* (he died in 1955), this step is not factual if the query explicitly mentions that *Einstein died in 1955*. Apart from that, if the previous steps provide the premises for reaching 1955, *i.e. Einstein was born in 1879, and he died at the age of 76*, the step is invalid.

While the standard practice is to treat factuality and validity separately (Prasad et al., 2023; Zhu et al., 2024b; Jacovi et al., 2024), the boundary between stating facts and making logical inferences is often vague, especially in commonsense reasoning. For example, if the step states *Einstein died between 1960 and 1970* when given the information *Einstein died in 1955*, is this step a factual error or a logical error? The boundary heavily relies on the definition of what can be taken as granted, which is also a key factor in defining coherence. REVEAL (Jacovi et al., 2024) delegates the decision to human annotators, and shows that LLMs (Anil et al., 2023; Brown et al., 2020) perform poorly (F1<0.65) at classifying the steps between factual statements and logical inference.

Validity↔Coherence Existing works often treat coherence as a subtype of validity (Golovneva et al., 2023a; Zhu et al., 2024b; Kim et al., 2025a; Jacovi et al., 2024), as both criteria judge a step based on its previous steps. However, validity and coherence are different by definition, as validity focuses on the logical correctness of a step while coherence focuses on the pragmatic aspect of informativeness. For instance (Figure 3-Coherence), omitting a step (Step 3) from the correct trace will make the subsequent step (Step 3') incoherent, but it is still valid since it can be eventually deduced from the query and previous steps.

Validity↔Utility Previous studies have continuously pointed out that validity does not necessarily lead to utility and vice versa (Lyu et al., 2023; Nguyen et al., 2024). One case is *shortcut reasoning* (Schnitzler et al., 2024; Lee and Hwang, 2025), where LLM generates invalid Chain-of-thoughts but guesses the correct answer directly from the query. ProcessBench (Zheng et al., 2024a) reports that invalid traces with correct answers can be easily found in challenging problems, reaching 51.8% in the olympiad-level Omni-MATH (Gao et al., 2024b).

The distinction between validity and utility has been highlighted by multiple empirical results. Treating these metrics as different yields substantial performance gain when training sequence classifiers (Zhang et al., 2025) and in Best-of-N decoding

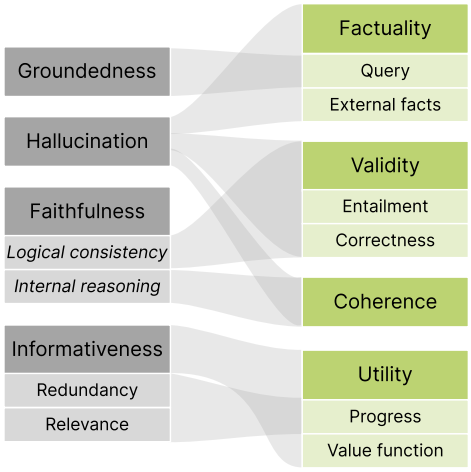

Figure 5: A Sankey diagram displaying the relationship between commonly used terminologies (left) to the proposed taxonomy (right).

(Sun et al., 2024; Kim et al., 2025b). See Section 6 for details.

C.2 Comparison to other definitions

Hallucination is most commonly defined as *"models either generating (1) nonsensical or (2) unfaithful to the source content"* (Ji et al., 2023; Banerjee et al., 2024; Huang et al., 2024b), which corresponds to (1) validity/coherence and (2) factuality. However, some works restrict the meaning of hallucination to factual errors, *i.e. "models generating description tokens that are not supported by the source inputs"* (Xiao and Wang, 2021; Akbar et al., 2024).

Faithfulness is also used with different senses. The most common definition for faithfulness is *"logical consistency between the generated text and the query/previous steps"* (Maynez et al., 2020; Creswell and Shanahan, 2022; Huang et al., 2024b), which includes both factuality (query groundedness) and validity (previous step). Instead, faithfulness can be used as *"accurately representing the model's internal reasoning process"* (Lyu et al., 2023; Lanham et al., 2023). Under this definition, the final step containing the answer is unfaithful if it is not supported by the previous steps, which falls under the definition of coherence.

Informativeness is defined as "providing new information that is helpful towards deriving the generated answer" (Golovneva et al., 2023b; Prasad et al., 2023). Lack of informativeness is often described as **redundancy** *"removing the step does not affect the reasoning process"* (Chiang and Lee, 2024; Song et al., 2025; Zhou et al., 2024) or **irrelevance** *"unrelated to the query's topic or task"* (Wang et al., 2023a; Zhou et al., 2024; Jacovi et al., 2024). Informativeness is synonymous with utility, as it aims to evaluate the contribution of a step to reaching the final answer.

D Details for Section 6

This section provides further details regarding the Section 6, specifically Figure 4.

D.1 Estimating Compute

To estimate the compute in Figure 4, we follow the approximation equation from Snell et al. (2024); Kim et al. (2025b). Specifically, the computational cost can be asymptotically approximated as

$$C \in O(N \times L),$$

where C is the total computational cost, N is the number of parameters, and L is the number of tokens. Note that since all compared evaluators use the same base model, N remains constant.

Below, we describe how the computation budget for each method is calculated in Figure 4:

- Unit relative compute (1) corresponds to a single forward pass for an average-length trace. This applies to *Fine-tuned Sequence Classifiers*, as they take the whole trace as the input.
- The *Base Model*, *Fine-tuned Critic Models* (She et al., 2025), and *Fine-tuned LRMs* evaluate each step with a separate forward pass. Thus, the compute is scaled by the number of steps per trace, which is 6.11 on average in ProcessBench (GSM8k + MATH). Note that LRMs like `DeepSeek-Distill-Qwen-2.5-7B` (DeepSeek-AI, 2025) generate significantly longer traces, with L scaled by 7.58.
- *PARC* (Mukherjee et al., 2025) also uses step-wise critic evaluations, but only using the *Partial context* (average 1.57 premises per step) makes PARC require lower compute by reducing L.
- In *Majority Voting* setting, where 8 step-wise evaluations are sampled per step and aggregated via majority voting, the total computation cost is multiplied by 8.

D.2 Data source

- For *Base model* and *Majority voting* scores, authors conducted experiments with `Qwen-2.5-7B-Instruct` using the code from Kim et al. (2025b). *Fine-tuned LRM* scores are as reported in the same paper.
- *Sequence classifiers* scores are obtained from Zhang et al. (2025).
- *Partial context* scores are provided by the authors of PARC (Mukherjee et al., 2025), upon requested by the authors of this survey. While the currently available version of the paper does not contain the result, it will appear in the published version.
- *Fine-tuned Critic Model* scores are from She et al. (2025).