ReviewScore: Misinformed Peer Review Detection with Large Language Models

Anonymous ACL submission

Abstract

Peer review serves as a backbone of academic research, but in most AI conferences, the review quality is degrading as the number of submissions explodes. To reliably detect low-quality reviews, we define *misinformed* review points as either "weaknesses" in a review that contain incorrect premises, or "questions" in a review that can be already answered by the paper. We verify that 15.2% of weaknesses and 26.4% of questions are *misinformed* and introduce REVIEWSCORE indicating if a review point is *misinformed*. To evaluate the factuality of each premise of weaknesses, we propose an automated engine that reconstructs every *explicit* and *implicit premise* from a weakness. We build a human expert-annotated REVIEWSCORE dataset to check the ability of LLMs to automate REVIEWSCORE evaluation. Then, we measure human-model agreements on REVIEWSCORE using eight current state-of-the-art LLMs. The models show F1 scores of 0.4–0.5 and kappa scores of 0.3–0.4, indicating moderate agreement but also suggesting that fully automating the evaluation remains challenging. A thorough disagreement analysis reveals that most errors are due to models' incorrect reasoning. We also prove that evaluating premise-level factuality shows significantly higher agreements than evaluating weakness-level factuality.[1]

1 Introduction

Peer review is an evaluation of academic work by experts to assess its originality, significance, and validity before publication (Kelly et al., 2014; Taylor and Francis, 2025). In AI conferences, as the number of submissions is exponentially increasing, required number of reviewers is also exploding. As a result, review quality is degraded, which undermines the integrity and reliability of a peer-review system (Stelmakh et al., 2021; Cortes and Lawrence, 2021; Shah, 2022; Kim et al., 2025). Due to the importance of the issue, previous works propose criteria for evaluating review quality. However, we observe a trade-off between applicability to reviews and specificity of rubrics. Goldberg et al. (2025) and Du et al. (2024) proposes criteria that could apply to nearly every review but those are quite vague and subjective. Guo et al. (2023), Sadallah et al. (2025), Purkayastha et al. (2025), and Ou et al. (2025) introduce specific and objective criteria but those target narrow scope of reviews.

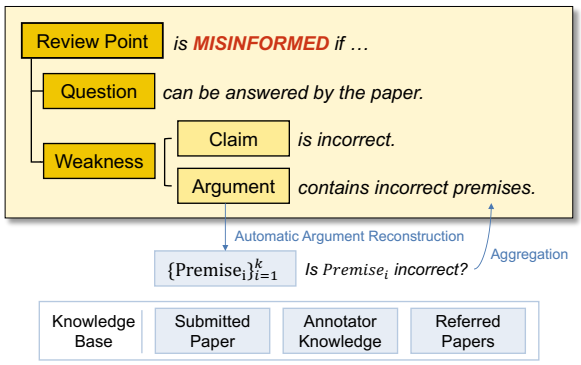

Figure 1: Overview of REVIEWSCORE. Each review point in a review is categorized into question and weakness. We further categorize weakness into claim and argument by the presence of supporting reasons. Based on an appropriate knowledge base, if a question is answerable by the paper, a claim is factually incorrect, or an argument contains factually incorrect premises, then the review point is *misinformed*. For arguments, to extract all *explicit* and *implicit premises*, we also introduce an automatic argument reconstruction engine.

To resolve this issue, we introduce two specific yet applicable criteria of a review quality: *unanswerability of questions* and *factuality of weaknesses*. To select the criteria, we recruited a group of human experts and let them independently analyze a small subset of ICLR reviews[2]. Specifically,

[1] We will make the source code and dataset publicly available.

[2] A group of three graduate students studying AI analyze reviews of ICLR submissions in OpenReview. Detailed process of this group work is described in Appendix B.

each of them decomposed a review into *review points*, which are formally defined as follows.

Definition 1 (Review Point)**.** A review point is a single, self-contained unit of evaluation or inquiry in a review–either a weakness or a question–that stands on its own semantically.

Each human evaluated quality of review points, and they discussed trustworthy criteria to detect low-quality. Based on the discussion, we formally define a *misinformed* review point as follows.

Definition 2 (Misinformed Review Point)**.** A review point is *misinformed* if and only if

- a question stated in a review can be already answered by the paper, or
- a weakness stated in a review is incorrect or contains incorrect premises regarding the paper.

In this work, the human annotation shows that 26.4% of questions and 15.2% of weaknesses are *misinformed*, which means that the current AI conference reviews contain considerable amount of *misinformed* review points.[3] We also note that these criteria aligns with reviewer guidelines of major AI conferences. For instance, ACL 2023 Peer Review Policies indicate that "before writing a negative review, check whether your questions are already answered." (Boyd-Graber et al., 2023), and NeurIPS 2025 reviewer guidelines indicate that "reviewers should minimize the chance of misunderstandings during the reviewing process" (NeurIPS 2025 Program Chairs, 2025).

Based on these observations, we define REVIEWSCORE that indicates if a review point is *misinformed* (Section 3.1). First, we define BASE REVIEWSCORE by directly applying the definition of *misinformed* review points in a 5-point scale. However, we discovered human-annotated factuality often diverges. This is because a weakness often contains both correct and incorrect *explicit premises* or incorrect *implicit premises*, which hinders humans to reliably annotate factuality. To resolve this issue, we further define ADVANCED REVIEWSCORE as an aggregation of premise factuality scores. If a weakness has no supporting reason, we call it as a *claim*, and if a weakness consists of more than one premises, we call it as an *argument*. We adopt two aggregation methods, logical conjunction, following the literature of logic (Beall et al., 2024), and weighted average, maintaining a 5-point scale (Figure 1). Before evaluating ADVANCED REVIEWSCORE, we have to extract all *explicit* and *implicit premises* from an argument. This process is called *argument reconstruction* in logic and critical thinking (Brun and Betz, 2016; Groarke, 2024; Dowden, 2024), and we construct an engine that automatically does this process (Section 3.2). To evaluate the ability of LLMs for detecting *misinformed* review points, we construct a human expert-annotated REVIEWSCORE dataset based on ICLR reviews (Section 3.3). We recruited 15 experienced graduate students and they dedicated total 244 hours for trustworthy human annotation.

We examine the reliability of automatic evaluation of REVIEWSCORE by measuring human-model agreements (Section 4). We use eight current state-of-the-art LLMs, including five proprietary and three open-sourced models, and the results show that there are only moderate agreements with the human experts (i.e., 0.4–0.5 F1 Scores and 0.3–0.4 Kappa Scores). A thorough human-model disagreement analysis reveals that models sometimes misunderstand the meaning of review points or predict scores that minority human annotators give. Furthermore, ADVANCED REVIEWSCORE clearly outperforms BASE REVIEWSCORE, which proves the effectiveness of premise-level factuality scoring.

To summarize our contributions:

1. We introduce REVIEWSCORE, a novel evaluation criteria that detects *misinformed* review points (i.e., questions that can be answered by the paper or weaknesses with incorrect premises).

2. To evaluate the factuality of premises, we propose an automatic argument reconstruction engine that generates a *valid* and *faithful* set of premises and conclusion.

3. We construct a trustworthy human expert-annotated dataset to measure the reliability of automatic evaluation of REVIEWSCORE.

4. We validate that an automatic REVIEWSCORE evaluation with current state-of-the-art LLMs only shows moderate agreement with human experts and conduct a comprehensive human-model disagreement analysis.

[3] A group of 15 graduate students annotate reviews of 40 works submitted to ICLR 2021-2023. Detailed process of this human annotation is described in Section 3.3.

	Target Unit	Groundedness	Factuality	Premise Factuality	(Un)Answerability
Shin et al. (2025)	Full Review	×	×	×	×
Purkayastha et al. (2025)	Weakness	×	×	×	×
Guo et al. (2023)	Full Review	✓	×	×	×
Sadallah et al. (2025)	Weakness	✓	×	×	×
Ou et al. (2025)	Weakness	✓	×	×	×
Du et al. (2024)	Weakness Sentence	✓	✓	×	×
REVIEWSCORE (Ours)	Review Point	✓	✓	✓	✓

Table 1: Comparison of automatic review evaluation methods in five criteria. Full Review contains a set of strengths and weaknesses.

2 Related Works

Peer review evaluation. Previous works studied how to evaluate the quality of peer reviews. Shin et al. (2025) and Purkayastha et al. (2025) map reviews to a predefined set of review types (i.e., types of facets and lazy thinking, respectively). However, they do not cover groundedness or factuality of reviews, which are emphasized as *understanding* and *substantiation* in Goldberg et al. (2025). Guo et al. (2023), Sadallah et al. (2025), and Ou et al. (2025) evaluate reviews by deciding whether the reviews are grounded and supported by a target paper or mapping reviews to a set of claims of the paper. The downside is that they do not explicitly evaluate whether the reviews are factually correct or not, leaving it as a future work. Du et al. (2024) go one step further and evaluate factuality of every sentence in weaknesses. However, the main drawback is that a sentence-level factuality cannot fully capture correctness of an underlying logic of weaknesses. Our work resolves this issue by first reconstructing a weakness into a set of explicit and implicit premises and then evaluating factuality of each premise. Moreover, our work also explicitly evaluates a quality of questions, whereas the aforementioned prior works mostly focus on evaluating weaknesses or a full review. These comparisons are summarized in Table 1.

Argument evaluation. In logic and critical thinking, an argument is a list of statements, one of which is the conclusion and the others are the premises (Dutilh Novaes, 2022; Lau and Chan, 2025). To evaluate an argument, we need to follow two steps. First, we have to identify and reconstruct the argument into a set of premises and conclusion, which is called an argument reconstruction (Brun and Betz, 2016; Groarke, 2024; Dowden, 2024). Then, we evaluate whether each premise is factually correct. An argument reconstruction should both be *valid*, which means premises deductively imply a conclusion, and *faithful*, which means premises and a conclusion accurately and completely represents an original argument (Brun and Betz, 2016; Betz and Richardson, 2021). Previously, Betz and Richardson (2021) trains a T5 model for argument reconstruction. However, the training datasets are either synthetic or polished and the reconstruction do not require any additional context information. In contrast, our work targets peer reviews, which include real-world unpolished arguments, and the reconstruction requires an entire paper to fully understand the context of arguments.

3 ReviewScore

We newly define REVIEWSCORE that measures how *misinformed* a review point is (Section 3.1). To evaluate REVIEWSCORE, we also introduce an engine that automatically extracts every *explicit* and *implicit premises* from a weakness (Section 3.2). Lastly, we construct a human expert-annotated dataset that evaluates LLM's ability to evaluate REVIEWSCORE (Section 3.3).

3.1 Definition

Our goal of defining REVIEWSCORE is to detect *misinformed* review points. Following this goal and the review quality criteria discussion in Section 1, we first define BASE REVIEWSCORE as *factuality of weaknesses* and *unanswerability of questions*. The following definition formally describes it.

Definition 3 (BASE REVIEWSCORE)**.** Let x be a review point (either a weakness or a question) about a submitted paper S. Define

$$\text{Factuality}_S : \mathcal{W} \to \{1,2,3,4,5\},$$
$$\text{Unanswerability}_S : \mathcal{Q} \to \{1,2,3,4,5\},$$

where $\mathcal{W}$ and $\mathcal{Q}$ are, respectively, the sets of weaknesses and questions appearing in a review of S[4]. We considered score 1–2 as *Misinformed* and score 3–5 as *Not misinformed* for binary classification

[4]Detailed rubric is described in Appendix H.

setup. The BASE REVIEWSCORE of x is:

$$\mathrm{ReviewScore}_{\mathrm{base}}(x) = \begin{cases} \mathrm{Factuality}_S(x) & \text{if } x \in \mathcal{W}, \\ \mathrm{Unanswerability}_S(x) & \text{if } x \in \mathcal{Q}. \end{cases}$$

For simplicity, we call the BASE REVIEWSCORE of weakness and questions as WScore and QScore.

However, during the group discussion in Section 1, we discovered that human annotators' evaluations on factuality diverge if a weakness contains both factual and nonfactual premises or a nonfactual premise is implicitly presumed. It happens since the human annotators implicitly weigh the importance of underlying premises of a weakness and then decide the final factuality score.

To resolve this issue, we further define ADVANCED REVIEWSCORE. We categorize weaknesses into arguments and claims based on whether there are supporting reasons or not. We keep the definition of WScore to evaluate claims, but further develops a finer-grained score to evaluate arguments. Following the literature of critical thinking (Brun and Betz, 2016; Groarke, 2024; Dowden, 2024), we reconstruct an argument into a premise-conclusion structure and then define ADVANCED REVIEWSCORE for arguments as (an aggregation of) *factuality of premises*. The following definition formally describes it.

Definition 4 (ADVANCED REVIEWSCORE)**.** Let x be a review point about a submitted paper S. Let $\mathcal{C}$, $\mathcal{A}$, and $\mathcal{Q}$ denote, respectively, the sets of claims, arguments, and questions in a review of S. For $x \in \mathcal{A}$, let its (*explicit* and *implicit*) premises be $\mathcal{P}(x) = \{p_1, \ldots, p_k\}$ with $\{p_1, \ldots, p_k\} \vdash C$ for the conclusion C of x. Let $\mathcal{K}$ be the set of knowledge bases available for factuality judgments (i.e., S, annotator knowledge, referred papers), and let

$$\mathrm{Factuality} : \mathcal{U} \times \mathcal{K} \to \{1, 2, 3, 4, 5\},$$
$$\mathcal{U} := \mathcal{C} \cup \bigcup_{x \in \mathcal{A}} \mathcal{P}(x),$$

be a 5-point scoring function for claims and premises given a knowledge base. Define a selector $\mathrm{KB} : \mathcal{U} \to \mathcal{K}$ that chooses the knowledge base used for each item (for claims $x \in \mathcal{C}$, $\mathrm{KB}(x) = S$; for premises p_i, $\mathrm{KB}(p_i) = \mathrm{KB}_i \in \mathcal{K}$). For a given $\mathrm{KB}(x) \in \mathcal{K}$, we notate the factuality function as $\mathrm{Factuality}_{\mathrm{KB}(x)}(\cdot)$. Agg is an operator that aggregates a list of scores to a single score. The ADVANCED REVIEWSCORE of x is:

$$\mathrm{ReviewScore}_{\mathrm{adv}}(x) = \begin{cases} \mathrm{Factuality}_S(x) & \text{if } x \in \mathcal{C}\,, \\ \mathrm{Agg}\big(\mathrm{Factuality}_{\mathrm{KB}_i}(p_i)_{p_i \in \mathcal{P}(x)}\big) & \text{if } x \in \mathcal{A}\,, \\ \mathrm{Unanswerability}_S(x) & \text{if } x \in \mathcal{Q}\,. \end{cases}$$

For simplicity, we call the ADVANCED REVIEWSCORE of claims, arguments, and questions as ClaimScore, ArgScore, and QScore.

Aggregation methods. To aggregate premise factuality scores as a single ArgScore, we introduce two aggregation methods: logical conjunction and weighted average. Following the literature of logic, an argument is *true* if and only if all premises are *true*. We define a premise is *true* if and only if it has factuality score 3–5, and otherwise, it is *false*. We dubbed this aggregation as logical conjunction, which follows the binary classification setup. However, if an annotator mistakenly evaluates one of premises, then the error propagates to the entire argument. To alleviate this issue, we also aggregate by a weighted average. Since it is difficult to measure the importance of premises, we instead weighted scores by untrivialness of premises (with a scale of 0–2). This is intended to simply filter out trivially true premises by measuring their importance as 0.[5] Specifically, (un)trivialness is decided based on the common knowledge of CS/AI-majoring undergrad students.

3.2 Automatic Argument Reconstruction

To evaluate ArgScore, we have to extract (*explicit* and *implicit*) premises $\mathcal{P}(x)$ from an argument $x \in \mathcal{A}$. Since human experts require significant amount of time and costs to do this, we automate it using LLMs. First, we check if a model could directly reconstruct an argument by giving detailed instructions. To preserve the context of an argument, we also give the model a submitted paper S. However, it mostly fails to generate *valid* and *faithful* reconstructions[6]. To resolve this issue, we

[5] Since we reconstruct every argument as valid (i.e., a set of premises deductively implies a conclusion), there are often conditional premises that make an argument valid but are trivially true. Detailed rubric of untrivialness is described in Appendix H.

[6] Detailed numerical results and qualitative analysis are reported in Appendix C.

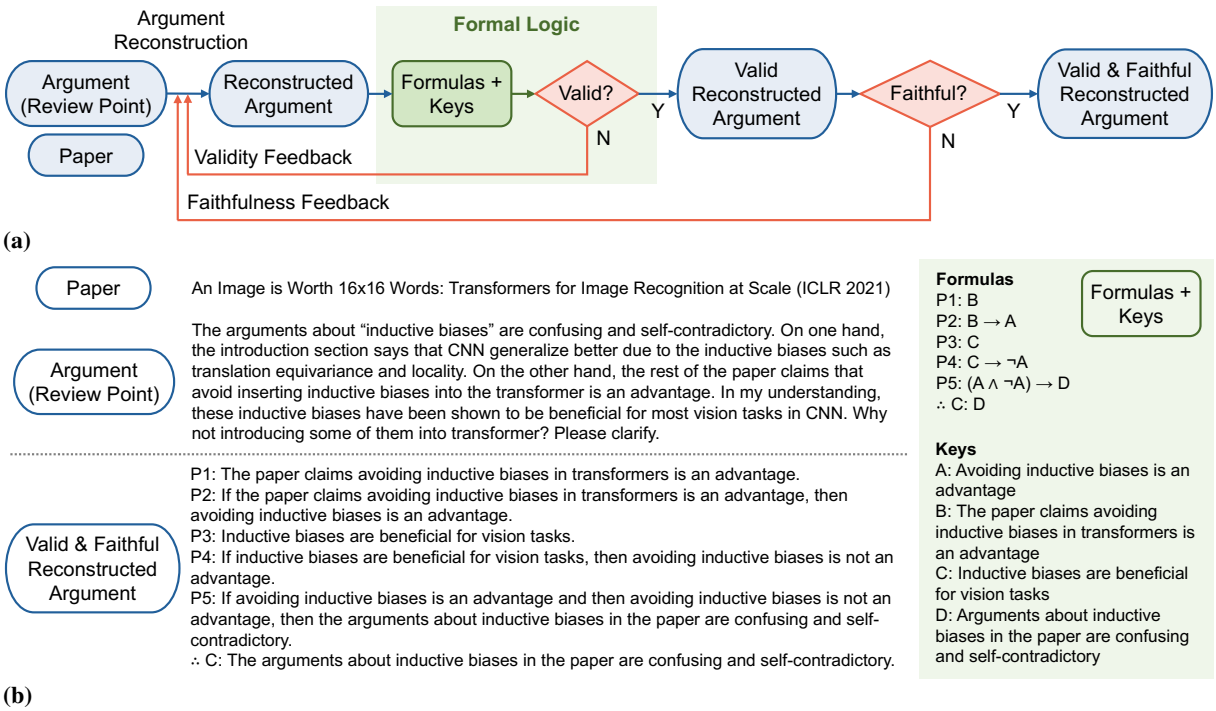

Figure 2: (a) Overview of an automatic argument reconstruction. Given an argumentative review point with a paper, a model first generates a reconstructed argument (i.e., a set of premises and conclusion). To check its *validity*, a model translates a NL reconstructed argument into FOL formulas, and then a SAT solver judges if it is *valid*. To check its *faithfulness*, a model translate FOL formulas back into the NL domain, and a model judges if the reconstruction is *faithful*. If one of two criteria does not met, then corresponding NL feedback is given to the generator model. (b) A representative example. We sample a review point of (Dosovitskiy et al., 2020) and its reconstruction along with corresponding formulas and keys.

add two feedback loops to ensure *validity* and *faithfulness* of an argument reconstruction (Figure 2a).

An LLM alone often falls short in ensuring *validity*, so we include a SAT solver which could automatically judges the *validity* of a set of premises and a conclusion without any logical errors. To do that, an LLM translates a set of natural language (NL) premises and conclusion into first-order logic (FOL) formulas. Then, a SAT solver decides whether the premises deductively implies the conclusion. If the reconstruction is *valid*, then an LLM translates formalized premises and conclusion back into NL domain. This process is called *logical streamlining*, which means to rephrase NL premises or conclusion in order to make their logico-semantic structure more transparent (Bowell and Kemp, 2014; Brun and Betz, 2016; Betz and Richardson, 2021). We then pass these *streamlined* NL premises and conclusion to the subsequent faithfulness feedback loop. However, if the reconstruction is *invalid*, then we feed a rule-based reward signal to the argument reconstructor to regenerate the reconstruction. There are two types of reward signals, one is a naive signal that says the formalized premises do not imply the conclusion, and the other one tells that the proof is circular.

Although the reconstruction is now *valid*, it should also *faithfully* represents the original argument. To ensure that, we prompt an LLM to judge whether the reconstruction is *faithful* or not and justify its decision. If the reconstruction is *faithful*, then we stop iterating the loop. If the reconstruction is *unfaithful*, then we feed LLM judge's justification to the argument reconstructor to regenerate the reconstruction. To minimize model calls in practice, we connect these two loops in series so that only *valid* reconstructions are judged for their *faithfulness*. Also, we limit the total number of loop iteration to 10 and return the last reconstructed argument if the loop fails to generate *valid* and *faithful* reconstruction. We provide an example reconstruction in Figure 2b, and details on feedback loops and model prompts are described in Appendix C and H, respectively.

Quality of argument reconstruction. We measure *validity* and *faithfulness* of reconstructed arguments using a SAT solver and human annotators, respectively. With Claude Sonnet 3.7 as a base LLM, every reconstruction is *valid*, and average *faithfulness* score is 4.5 / 5. Detailed analysis including comparison with direct reconstruction is

described in Appendix C.

3.3 Dataset Construction

Our final destination of proposing REVIEWSCORE is to *automatically* filter out *misinformed* review points using LLMs. To verify LLM's ability to do that, we build a human expert-annotated dataset to measure an agreement between humans and LLMs on REVIEWSCORE evaluation. Our dataset contains total 657 annotated review points, consisting of 143 questions, 92 claims, and 422 arguments. Specifically, 1,748 premises of the arguments are manually annotated [7].

Human-review matching. We recruit 15 graduate students studying AI as human annotators, and they annotate total 40 papers submitted to ICLR 2021–2023[8]. Specifically, we first make five groups by their research interests. For each group, three human annotators discuss which papers to annotate and select eight papers that are relevant with all three. Then, each human annotates selected eight papers' review points which are preprocessed from OpenReview.

Data curation process. We collaboratively use an LLM and humans for REVIEWSCORE data curation, where an LLM preprocesses reviews and then humans annotate those. Given a review, an LLM extracts independent review points. For each review point, an LLM automatically annotates the type (i.e., claim, argument, or question) and human verifies it. If the review point is a question, then human scores if the question is (un)answerable by the paper in a 5-point scale and justifies it if needed. If the review point is a claim, then human scores if the claim is true in a 5-point scale and justifies it if needed. If the review point is an argument, then human scores the argument's factuality same as in claims (i.e., BASE REVIEWSCORE). To annotate ADVANCED REVIEWSCORE for arguments, we first run automatic argument reconstruction engine (Section 3.2) to extract underlying premises of the argument. After that, human scores if the reconstruction is *faithful* in a 5-point scale. If the *faithfulness* score is less than 4 (i.e., faithful, but one or two minor changes recommended), then they skip the subsequent annotations. Otherwise, they judge the factuality of premises. For each premise, they first select a knowledge base (i.e., submitted paper, annotator knowledge, or referred papers), score if the premise is true based on the knowledge base in a 5-point scale, and score if the premise is (un)trivial in a 3-point scale. They justify any of three decisions if needed.

Trustworthiness of human annotation. We provide annotators detailed guidelines and an hour-long online orientation session. During annotation, they are allowed to use any related materials or tools (e.g., discussion between authors and reviewers in OpenReview, web search, etc.). Furthermore, we highly encourage the annotators to communicate with us through a group chat so that we could give them instant responses to their questions and share with all, which builds a global consensus among human annotators.

To reduce annotator bias, three humans first independently annotate review points. To control label quality, we further conduct a disagreement-aware recheck where humans are given disagreed instances[9] and revise these only if clear mistakes are indicated. Note that humans independently perform this recheck without having access to others' labels. Thanks to these efforts, despite the difficulty of the work, we obtain median 0.489 and highest 0.663 inter-annotator agreement in Krippendorff's Alpha (Hughes, 2021) across annotator groups.

However, there still exists few disagreements after the recheck. To resolve these, we conduct an additional group discussion to finalize human labels of the disagreed instances. We basically take median values from three annotations as final human labels, whereas for few disagreed instances, we exceptionally take post-discussion labels as final ones. Details are described in Appendix E.

4 Reliability of Automatic Evaluation of ReviewScore

To evaluate LLM's ability to evaluate REVIEWSCORE, we describe an experimental setup (Section 4.1) and show human-model agreement results (Section 4.2). We also compare the effectiveness of BASE and ADVANCED REVIEWSCORE (Section 4.3). We further analyze human-model disagreements and the effect of providing authors response to models (Section 4.4).

[7] Details of the dataset construction and statistics are described in Appendix D.

[8] Since ICLR 2024–2025 submission drafts are not opened to public, we exclude these years.

[9] Annotations are disagreed if a difference of maximum and minimum scores is larger than or equal to 2.

Table 2: Human-model agreement on REVIEWSCORE evaluation.

Model	ClaimScore		ArgScore		QScore		REVIEWSCORE	
	F1	Kappa	F1	Kappa	F1	Kappa	F1	Kappa
Proprietary models								
Claude Sonnet 3.7	0.160	0.156	0.462	0.261	0.576	0.410	0.462	0.336
Claude Sonnet 4	0.222	0.165	0.403	0.272	**0.579**	**0.425**	0.448	**0.378**
GPT-4o	0.000	0.093	0.333	0.244	0.476	0.291	0.385	0.347
GPT-5	0.125	0.024	**0.481**	0.353	0.560	0.340	**0.482**	0.341
Gemini 2.5 Flash	**0.240**	**0.169**	0.466	**0.402**	0.522	0.265	0.464	0.357
Open-sourced models								
Qwen3-235B-A22B	**0.231**	0.142	**0.413**	0.148	**0.544**	0.243	**0.452**	0.262
Llama 3.3	0.118	0.097	0.338	0.108	0.514	**0.254**	0.415	**0.311**
DeepSeek-V3	0.000	**0.180**	0.298	**0.176**	0.530	0.192	0.383	0.301

4.1 Setup

Given score rubrics in a 5-point scale, an LLM evaluates REVIEWSCORE according to a knowledge base it selects. We only provide a submitted paper to a model since we assume that the model has a human-level internal knowledge and has a general understanding of referred papers. Detailed model prompts are described in Appendix H.

Language models. To measure LLM's ability to automatically evaluate REVIEWSCORE, we perform experiments on eight current state-of-the-art LLMs that achieve significantly high alignments with humans. We include five proprietary models, Claude Sonnet 3.7 (Anthropic, 2025a), Claude Sonnet 4 (Anthropic, 2025b), GPT-4o (OpenAI, 2024), GPT-5 (OpenAI, 2025), and Gemini 2.5 Flash (Gemini Team, Google, 2025), and three open-sourced models, Qwen3-235B-A22B (Yang et al., 2025), DeepSeek-V3 (DeepSeek-AI et al., 2024), and Llama 3.3 (AI@Meta, 2024)[10]. To get consistent and reliable scores from LLM judges, we set a low temperature (i.e., 0) and select the highest probability response from the model (Liang et al., 2022; Liu et al., 2023; Gu et al., 2024). We exclude reasoning models as LLM judges since REVIEWSCORE mostly depends on grounding and evidence, not longer or smarter chains of thought.

Evaluation metrics. We use different sets of metrics for two types of problem formulations, binary classification and 5-point scale scoring. For the binary classification, since majority of human-annotated labels are *Not misinformed*, we mainly use F1 score which is robust to class imbalance. For the 5-point scale scoring, since majority of human-annotated scores are 4 and 5, we mainly use Quadratic Weighted Kappa (Warrens, 2012), a variant of Cohen's Kappa (Cohen, 1960), that is robust to the skewed data distribution. Hereinafter, we call this metric Kappa for simplicity. To provide more comprehensive results, we additionally use Precision and Recall for the binary classification and Pearson rank correlation and Gwet's AC2 (Gwet, 2001) for the 5-point scale scoring.

[10]Details of model specifications are described in Appendix G.

4.2 Main Results

We empirically validate the alignment of human-annotated and model-estimated REVIEWSCORE using different models and evaluation metrics in Table 2. Most models show 0.4–0.5 F1 score and 0.3–0.4 Kappa score, which indicates moderate agreement between humans and models on REVIEWSCORE evaluation. However, there are differences in human-model agreements for three types of review points. Regardless of the models, questions show the highest agreement, arguments follow subsequently, and claims show the lowest agreement. Specifically, for claims, some models show zero F1 score or nearly zero Kappa score. We analyze human-model disagreements thoroughly in Section 4.4 and conclude that since claims lack supporting evidence and are often value-laden, models often misinterpret the intended meaning of the claims or judge differently than humans. More results and qualitative analysis of model evaluation are described in Appendix F.

4.3 Base vs. Advanced ReviewScore

We compare human-model agreements of BASE and ADVANCED REVIEWSCORE using different models and evaluation metrics. Since the only difference is in defining scores for arguments, we compare REVIEWSCORE for arguments in Table 3. To verify the effectiveness of aggregation methods, we additionally include ADVANCED REVIEWSCORE for arguments without aggregation (i.e., factuality scores of premises). Regardless of models, ADVANCED REVIEWSCORE clearly shows higher

Table 3: Comparison of human-model agreement of BASE vs. ADVANCED REVIEWSCORE.

Model	Metric	Base	Advanced w/o Agg	Advanced
Claude Sonnet 3.7	F1	0.157	0.333	**0.462**
	Kappa	0.120	**0.308**	0.261
GPT-5	F1	0.218	0.370	**0.481**
	Kappa	0.182	**0.366**	0.353
Gemini 2.5 Flash	F1	0.137	0.395	**0.466**
	Kappa	0.124	0.386	**0.402**
DeepSeek-V3	F1	0.146	0.167	**0.298**
	Kappa	0.091	**0.193**	0.176

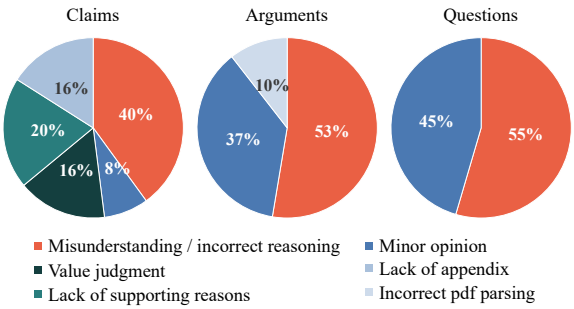

Figure 3: Types of human-model disagreements.

agreements than BASE REVIEWSCORE. For Gemini 2.5 Flash, ADVANCED REVIEWSCORE performs 3.40× higher F1 score and 3.24× higher Kappa score than the BASE REVIEWSCORE. For the aggregation methods, logical conjunction contributes to higher agreements (i.e., F1 scores are consistently higher with aggregation). However, weighted average shows marginal improvements or slight degradations. We manually investigate human-model disagreements and observe that humans and models have slightly misaligned criteria on whether a given premise is trivially true or not based on the common knowledge of CS/AI-majoring undergrad students.

4.4 Analysis

Human-model disagreements. We analyze types of human-model disagreements in Figure 3. Across all review points, there are two common types of disagreements, which are models' misunderstanding or incorrect reasoning and models' predictions which correspond to minority of human-annotated scores. These two types comprise nearly all disagreements on arguments and questions, and about half of disagreements on claims. For claims, we observe that 36% of disagreements are caused by either claims are value-laden or lack supporting reason. These make the factuality judgment of claims subjective, leading to a low human-model agreement. Practical limitations such as incorrect pdf parsing and not providing appendix to models cause nontrivial portion of disagreements.

Table 4: Effect of providing Authors Response (AR) to a model for REVIEWSCORE evaluation.

	F1		Kappa	
	w/o AR	w/ AR	w/o AR	w/ AR
WScore	0.194	**0.243**	0.149	**0.198**
ArgScore w/o Agg	0.315	**0.385**	0.322	**0.372**
ArgScore	0.403	**0.493**	0.272	**0.338**
QScore	**0.579**	**0.578**	**0.425**	0.410
ClaimScore	**0.222**	0.125	0.165	**0.221**
REVIEWSCORE	0.448	**0.498**	0.378	**0.416**

Helpfulness of authors response. We study whether providing authors responses of reviews to a model benefits the automatic REVIEWSCORE evaluation using Claude Sonnet 4 in Table 4. Overall, providing authors response leads to higher human-model agreement of REVIEWSCORE evaluation. However, we observe that providing authors response has marginal or sometimes negative effect on QScore and ClaimScore evaluation. These observations show that authors response largely benefits argument or premise factuality evaluation by providing additional cues for model judgments, whereas does not benefit question (un)answerability and claim factuality by injecting authors bias.

5 Conclusion

We introduce REVIEWSCORE, a new evaluation metric of peer review quality, focusing on detecting *misinformed* review points. To evaluate LLM's ability for REVIEWSCORE evaluation, we also construct a trustworthy human-annotated dataset. The results show a moderate human-model agreement, and further comprehensive disagreement analysis reveals that current state-of-the-art LLMs sometimes misunderstand or reason incorrectly. However, we confirm that premise-level factuality shows significantly higher human-model agreements than weakness-level factuality, which proves the effectiveness of our method.

Limitations

We acknowledge three types of limitations of our work.

First, there are practical limitations in the automated REVIEWSCORE evaluation. To save API calling costs, when the paper is given to LLMs, we only provide text and tables of the main paper, where any figure or appendix is excluded. We also observe nontrivial amount of incorrect pdf parsing.

Second, there are technical limitations in the automatic argument reconstruction. The reconstruction output is not always perfect since it depends on the capability of the base model. However, we observe a considerable output quality improvement when we upgrade the base model to Claude Sonnet 4, indicating that the reconstruction engine would perform better as the model improves in general.

Lastly, there are practical limitations in the dataset construction. Since human annotators are graduate students with varying skillfulness and the annotation requires significant amount of cognitive load, there are unavoidable noise in human annotation. Furthermore, a manual selection of papers by human annotators might introduce any kind of unintended biases. However, to collect the most reliable annotation under our limited budget, we inevitably choose this method.

References

AI@Meta. 2024. The llama 3 herd of models. Family report for Llama 3; used to cite Llama 3.3 70B Instruct.

Anthropic. 2025a. Claude 3.7 sonnet system card. Technical report, Anthropic PBC. System card / technical report.

Anthropic. 2025b. System card: Claude opus 4 & claude sonnet 4. Technical report, Anthropic PBC. System card / technical report.

Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, and Christopher Re. 2023. Ask me anything: A simple strategy for prompting language models. In *The Eleventh International Conference on Learning Representations*.

Jc Beall, Greg Restall, and Gil Sagi. 2024. Logical Consequence. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Summer 2024 edition. Metaphysics Research Lab, Stanford University.

Gregor Betz and Kyle Richardson. 2021. Deepa2: A modular framework for deep argument analysis with pretrained neural text2text language models. *arXiv preprint arXiv:2110.01509*.

Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2023. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*.

Tracy Bowell and Gary Kemp. 2014. *Critical Thinking: A Concise Guide*. Routledge.

Jordan Boyd-Graber, Naoaki Okazaki, and Anna Rogers. 2023. Acl'23 peer review policies. ACL 2023 Blog. Program Chairs blog post.

Georg Brun and Gregor Betz. 2016. Analysing practical argumentation. In *The argumentative turn in policy analysis: Reasoning about uncertainty*, pages 39–77. Springer.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Corinna Cortes and Neil D. Lawrence. 2021. Inconsistency in conference peer review: Revisiting the 2014 neurips experiment. *arXiv preprint arXiv:2109.09774*.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. 2023. Diffedit: Diffusion-based semantic image editing with mask guidance. In *The Eleventh International Conference on Learning Representations*.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2023. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *The Eleventh International Conference on Learning Representations*.

Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An efficient smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer.

DeepSeek-AI, Aixin Liu, Bei Feng, and 1 others. 2024. Deepseek-v3 technical report. Technical report; applicable to the V3-0324 checkpoint lineage.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Bradley H Dowden. 2024. Logical reasoning (dowden).

Jiangshu Du, Yibo Wang, Wenting Zhao, Zhongfen Deng, Shuaiqi Liu, Renze Lou, Henry Peng Zou, Pranav Narayanan Venkit, Nan Zhang, Mukund Srinath, and 1 others. 2024. Llms assist nlp researchers: Critique paper (meta-) reviewing. *arXiv preprint arXiv:2406.16253*.

Catarina Dutilh Novaes. 2022. Argument and Argumentation. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Fall 2022 edition. Metaphysics Research Lab, Stanford University.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. 2023. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*.

Gemini Team, Google. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. Technical report, Google DeepMind. Technical report; covers Gemini 2.5 Pro and 2.5 Flash.

Alexander Goldberg, Ivan Stelmakh, Kyunghyun Cho, Alice Oh, Alekh Agarwal, Danielle Belgrave, and Nihar B Shah. 2025. Peer reviews of peer reviews: A randomized controlled trial and other experiments. *PloS one*, 20(4):e0320444.

Leo Groarke. 2024. Informal Logic. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Spring 2024 edition. Metaphysics Research Lab, Stanford University.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*. Recommends temperature=0 to reduce randomness and stabilize judge outputs.

Yanzhu Guo, Guokan Shang, Virgile Rennard, Michalis Vazirgiannis, and Chloé Clavel. 2023. Automatic analysis of substantiation in scientific peer reviews. *arXiv preprint arXiv:2311.11967*.

Kilem Gwet. 2001. Handbook of inter-rater reliability. *Gaithersburg, MD: STATAXIS Publishing Company*, pages 223–246.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. 2023. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*.

Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. 2022. Language model compression with weighted low-rank factorization. In *International Conference on Learning Representations*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve.

John Hughes. 2021. krippendorffsalpha: An r package for measuring agreement using krippendorff's alpha coefficient. *arXiv preprint arXiv:2103.12170*.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun KIM, Stanley Jungkyu Choi, and Minjoon Seo. 2022. Towards continual knowledge learning of language models. In *International Conference on Learning Representations*.

Haozhe Ji, Pei Ke, Zhipeng Hu, Rongsheng Zhang, and Minlie Huang. 2023. Tailoring language generation models under total variation distance. In *The Eleventh International Conference on Learning Representations*.

Jacalyn Kelly, Tara Sadeghieh, and Khosrow Adeli. 2014. Peer review in scientific publications: benefits, critiques, & a survival guide. *Ejifcc*, 25(3):227.

Jaeho Kim, Yunseok Lee, and Seulki Lee. 2025. Position: The ai conference peer review crisis demands author feedback and reviewer rewards. *arXiv preprint arXiv:2505.04966*.

Yoonjeon Kim, Hyunsu Kim, Junho Kim, Yunjey Choi, and Eunho Yang. 2023. Learning input-agnostic manipulation directions in styleGAN with text guidance. In *The Eleventh International Conference on Learning Representations*.

Anna Kukleva, Moritz Böhle, Bernt Schiele, Hilde Kuehne, and Christian Rupprecht. 2023. Temperature schedules for self-supervised contrastive methods on long-tail data. In *The Eleventh International Conference on Learning Representations*.

Joe Lau and Jonathan Chan. 2025. What is an argument? Critical Thinking Web. Module A01: Argument analysis.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2022. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*.

Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2021. Mix{kd}: Towards efficient distillation of large-scale language models. In *International Conference on Learning Representations*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Standardizes eval settings; many HELM runs use temperature=0 for reproducibility.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. 2023. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2511–2522. Association for Computational Linguistics. Uses low temperature / deterministic decoding for stable evaluator outputs.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2023. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *The Eleventh International Conference on Learning Representations*.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2022. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*.

Jaehyun Nam, Jihoon Tack, Kyungmin Lee, Hankook Lee, and Jinwoo Shin. 2023. STUNT: Few-shot tabular learning with self-generated tasks from unlabeled tables. In *The Eleventh International Conference on Learning Representations*.

NeurIPS 2025 Program Chairs. 2025. Neurips 2025 reviewer guidelines. NeurIPS 2025 website. Guidelines page.

Ansong Ni, Jeevana Priya Inala, Chenglong Wang, Alex Polozov, Christopher Meek, Dragomir Radev, and Jianfeng Gao. 2023. Learning math reasoning from self-sampled correct and partially-correct solutions. In *The Eleventh International Conference on Learning Representations*.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*.

OpenAI. 2024. Gpt-4o system card. Technical report, OpenAI. System card / technical report.

OpenAI. 2025. Gpt-5 system card. Technical report, OpenAI. System card / technical report.

Jiefu Ou, William Gantt Walden, Kate Sanders, Zhengping Jiang, Kaiser Sun, Jeffrey Cheng, William Jurayj, Miriam Wanner, Shaobo Liang, Candice Morgan, and 1 others. 2025. Claimcheck: How grounded are llm critiques of scientific papers? *arXiv preprint arXiv:2503.21717*.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2023. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*.

Sukannya Purkayastha, Zhuang Li, Anne Lauscher, Lizhen Qu, and Iryna Gurevych. 2025. Lazyreview a dataset for uncovering lazy thinking in nlp peer reviews. *arXiv preprint arXiv:2504.11042*.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. In *The Eleventh International Conference on Learning Representations*.

Laura Eline Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. 2023. Large language models are not zero-shot communicators.

Abdelrahman Sadallah, Tim BaumgÃĪrtner, Iryna Gurevych, and Ted Briscoe. 2025. The good, the bad and the constructive: Automatically measuring peer review's utility for authors. *arXiv preprint arXiv:2509.04484*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, and 21 others. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Nihar B. Shah. 2022. Challenges, experiments, and computational solutions in peer review. *Communications of the ACM*, 65(6):76–87.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2022. How much can CLIP benefit vision-and-language tasks? In *International Conference on Learning Representations*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.

Hyungyu Shin, Jingyu Tang, Yoonjoo Lee, Nayoung Kim, Hyunseung Lim, Ji Yong Cho, Hwajung Hong, Moontae Lee, and Juho Kim. 2025. Mind the blind spots: A focus-level evaluation framework for llm reviews. *arXiv preprint arXiv:2502.17086*.

Satya Narayan Shukla and Benjamin Marlin. 2021. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*.

Satya Narayan Shukla and Benjamin Marlin. 2022. Heteroscedastic temporal variational autoencoder for irregularly sampled time series. In *International Conference on Learning Representations*.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Lee Boyd-Graber, and Lijuan Wang. 2023. Prompting GPT-3 to be reliable. In *The Eleventh International Conference on Learning Representations*.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. 2023. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*.

Ivan Stelmakh, Nihar B. Shah, Aarti Singh, and Hal Daumé III. 2021. A novice-reviewer experiment to address scarcity of qualified reviewers in large conferences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4785–4793.

Taylor and Francis. 2025. Understanding the peer review process. what is peer review? a guide for authors.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. 2023. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *The Eleventh International Conference on Learning Representations*.

Matthijs J Warrens. 2012. Some paradoxical results for the quadratically weighted kappa. *Psychometrika*, 77(2):315–323.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, and 1 others. 2025. Qwen3 technical report. Covers the Qwen3 family including the 235B A22B series (e.g., 2507 checkpoint).

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

Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. 2022. Graph-guided network for irregularly sampled multivariate time series. In *International Conference on Learning Representations*.

Yunhao Zhang and Junchi Yan. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

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

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023b. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

A Future Impact

We expect automatic REVIEWSCORE evaluation could greatly benefit different roles of the current peer reviewing system. For authors, by providing reconstruction of argumentative review points, it helps to understand or clarify the reviewer's intention and to formulate a rebuttal. For reviewers, by providing REVIEWSCORE of their review points, it allows them to verify the review quality by themselves and helps reviewers to better understand the paper. For metareviewers, by providing REVIEWSCORE of each reviewer, it could assist their final decisions. To summarize, it could serve as an automated system for managing the review quality.

B Review Quality Criteria Discussion

First, we recruit a group of three graduate students studying AI and NLP and let them independently analyze reviews of ten common manually selected submitted papers from ICLR 2021–2023[11]. Specifically, we guided the group to: decompose a review into several independent review points, evaluate quality of the review points by a 5-point scale with their own criteria, and justify their scores that ends with a meta-sentence that does not involve paper's context which is used for our further analysis. During human evaluation, we minimize our effort to provide detailed guidelines to facilitate bias-free human analysis.

Then, we categorize these justifications in order to find common features of low-quality review points. There are five common types: (1) questions that can be already addressed by the paper, (2) comments that reflect a misunderstanding of the paper, (3) out-of-scope remarks, (4) observations pointing out minor details, and (5) unclear points. However, for the last three types, we observe that majority of review points are agreed or argued by a single human annotator. In other words, given a review point, only a single human argues it is out-of-scope, whereas other two humans argue it is within scope and could be a potential drawback of the submitted paper. Following this pattern, a single human argues a review point is addressing minor details but others argue it is a major point, and a single human argues a review point is unclear whereas others do not agree with this. In contrast, the first two types of review points (i.e., answerable questions and misunderstood comments) are mostly agreed by two or more human annotators, indicating that these two types have more objective and trustworthy criteria for detecting low-quality reviews. Based on these observations, we focus on evaluating review points based on the first two criteria.

Lastly, we share some meaningful insights from this group discussion that strongly motivates our work. Based on the analysis of score justifications, given a review point, some (sub)sentences are high-quality or factually correct but others could be misinformed or factually incorrect. This later motivates our premise-level factuality evaluation (Section 3.1). Furthermore, during the group discussion, we observe that human annotators sometimes struggle which parts in a review point they should weigh more to evaluate the review point. This later motivates our aggregation methods which include logical conjunction and weighted average by untrivialness (Section 3.1). We also discover that human evaluations are sometimes incorrect. We leave this as a limitation of our work. However, to alleviate this issue, we recruit three human annotators for every instance and further ensure the annotation quality by providing them careful guidelines and actively communicating with them to build a global consensus on the evaluation criteria (Section 3.3).

C Automatic Argument Reconstruction Details and Results

C.1 Implementation details

We elaborate components of automatic argument reconstruction engine (Section 3.2, Figure 2) in the following. We also refer the corresponding prompt used in each step if it exists.

1. Given an argumentative review point, an LLM extracts a verbatim conjecture and its verbatim reason statements (Figure 23).

2. Given a verbatim conclusion and reasons of the argumentative review point and a corresponding paper parsed from a pdf file, an LLM reconstructs the argument into a premise-conclusion structure. At the same inference, the model also translates (or formalizes) NL premises and conclusion into corresponding FOL formulas and generate keys which assign NL meaning to variables and predicates. To

[11] Since those papers include papers written by authors of our work, we do not share the full list of target papers to keep anonymity. We will uncover the full list after the paper is published.

facilitate the model to generate a *valid* formalization, the model generates a deductive proof using formalized premises and conclusion at the end (Figure 24).

3. Given FOL premises and conclusion with keys and a deductive proof, an LLM extracts necessary FOL premises for the deductive proof, write a python program using Z3 (De Moura and Bjørner, 2008) that automatically checks the *validity* of the necessary FOL premises and conclusion, extracts a final FOL conclusion that is used in the python program, and judges whether the proof is circular (i.e., whether the final FOL conclusion is included in one of necessary FOL premises.) (Figure 25, Figure 26).

4. If the proof is circular, then a NL feedback indicating circularity of the proof is sent to Step 2, and the model re-generates an argument reconstruction. Otherwise, we run the python program that checks the *validity* of the reconstruction. However, if the program returns an error, the model takes this error message and re-generate the python program that fixes the error (Figure 27). If the reconstruction is *invalid*, then a NL feedback indicating *invalidity* of the reconstruction is sent to Step 2, and the model re-generates an argument reconstruction. Otherwise (i.e., if the reconstruction is *valid*), we proceed to the next step.

5. To check *faithfulness* of the reconstruction, the model first translates FOL premises and conclusion (one of the outputs of the model in Step 2) with keys (one of the outputs of the model in Step 1) into NL premises and conclusion (Figure 28). This process is called *logical streamlining* in logic and critical thinking (Bowell and Kemp, 2014; Brun and Betz, 2016; Betz and Richardson, 2021).

6. Lastly, given an original argumentative review point (or an argument) and the *streamlined* NL premises and conclusion, the model judges whether the reconstruction is *faithful* with justifications (Figure 29). If the reconstruction is *unfaithful*, then a NL feedback including the model's justifications is sent to Step 2, and the model re-generates an argument reconstruction accordingly. Otherwise, since the reconstruction is *valid* yet *faithful*, the feedback loop is finished and the *streamlined* NL premises and conclusion become a final argument reconstruction.

C.2 Quantitative Results

To verify effectiveness of the feedback loop, we report the argument reconstruction performance with and without feedback in Table 5. We mainly measure average *validity* and *faithfulness* of the reconstructed arguments using a SAT solver and human annotators, respectively. We provide score rubric for evaluating *faithfulness* in Figure 4. We also include the pass rate, which indicates whether the (last) reconstructed argument fulfill the *validity* and *faithfulness* criteria in the feedback loop. Furthermore, we report the average number of loop iterations to check if the feedback loops are actively used.

In Table 5, we verify that the proposed method (i.e., w/ feedback) achieves a perfect *validity* and nearly perfect *faithfulness* and pass rate, whereas the performance of the direct reconstruction (i.e., w/o feedback) largely lags behind that. We observe that the average number of loop iterations is 3.09, indicating active usage of feedback signals, but the number varies a lot depending on the clarity of logical structures of the arguments.

Table 5: Performance of Automatic Argument Reconstruction.

	w/o feedback	w/ feedback
Validity	0.895	**1.00**
Faithfulness	2.91	**4.47**
Pass Rate	0.369	**0.935**
# of loops	N/A	3.09 ± 2.73

Faithfulness Rubric for Argument Reconstruction

5: Definitely faithful, no change is essential
4: Faithful, but one or two minor details in premises need to change in order to fully expres s the original context
3: Faithful, but more than two minor details in premises need to change
2: Not faithful, one or two major error exist
1: Definitely not faithful, more than two major errors exist

Figure 4: Score rubric for evaluating *faithfulness* of argument reconstruction by human annotators.

C.3 Qualitative Results

We present three representative examples of model-generated argument reconstruction and compare outputs with and without the feedback loop. We

observe the direct reconstruction (i.e., w/o feedback) often generates either *invalid* or *unfaithful* reconstructed arguments, however, the proposed method (i.e., w/ feedback) generates *valid* yet *faithful* reconstructed arguments. We provide detailed analysis in Figure 5–7.

Input: Argument (Review Point) & Paper

Paper:
Least-to-Most Prompting Enables Complex Reasoning in Large Language Models (Zhou et al., 2023a)

Argument (Review Point):
The paper lacks insights on key questions. Does the model have ability to decompose questions very different from demo examples into correct sub-problems? Even the model solves each sub-problem correctly, what makes the model compose the final solution correctly?

Output: Reconstructed Argument

Direct Reconstruction (w/o feedback):
P1: The paper does not demonstrate whether the model has the ability to decompose questions very different from demo examples into correct sub-problems.
P2: The paper does not explain what makes the model compose the final solution correctly even when it solves each sub-problem correctly.
P3: If a paper fails to provide insights on key questions about a proposed method, then the paper lacks insights on key questions.
P4: Questions about decomposition ability and solution composition are key questions for the least-to-most prompting method.
∴ C: The paper lacks insights on key questions.

Proposed Method (w/ feedback):
P1: The paper does not demonstrate whether the model has the ability to decompose questions very different from demo examples into correct sub-problems.
P2: If the paper does not demonstrate whether the model has the ability to decompose questions very different from demo examples into correct sub-problems, then the paper fails to provide insights on the key question about decomposition ability.
P3: If the paper fails to provide insights on the key question about decomposition ability or fails to provide insights on the key question about solution composition, then the paper fails to provide insights on key questions.
P4: If the paper fails to provide insights on key questions, then the paper lacks insights on key questions.
∴ C: The paper lacks insights on key questions.

Analysis

- For the direct reconstruction, P1, P2, P4 semantically implies an intermediate conclusion that "The paper fails to provide insights on key questions about the proposed method.", but this connecting premise is not explicitly reconstructed. Therefore, this reconstruction is *invalid* [×]. However, if we consider that this connecting premise is implicitly presumed, then the reconstruction *faithfully* [✓] represents the original argument.
- For the proposed method, the reconstruction is logically *valid* [✓] yet *faithfully* [✓] represents the original argument.

Figure 5: Example #1 of automatic argument reconstruction.

Input: Argument (Review Point) & Paper

Paper:
ReAct: Synergizing Reasoning and Acting in Language Models (Yao et al., 2023)

Argument (Review Point):
ReAct alone works fairly well for some tasks, but self-consistent Chain of Thought often outperforms ReAct alone, and leads to fairly dramatic increases in performance. The best models use a smart combination of both CoT and ReAct, using heuristics that likely maximize performance on the dataset(s) (see 'ReAct → CoT–SC' strategy), which makes the performance of ReAct by itself slightly less impressive (though I still think this is a useful contribution for the community)

Output: Reconstructed Argument

Direct Reconstruction (w/o feedback):
P1: The best models use a smart combination of both CoT and ReAct.
P2: Models use heuristics that likely maximize performance on the dataset(s) (see 'ReAct → CoT–SC' strategy).
P3: If models use heuristics that likely maximize performance on the dataset(s), then the performance of ReAct by itself is less impressive.
P4: If the best models use a combination of both CoT and ReAct rather than ReAct alone, then the performance of ReAct by itself is less impressive.
P5: The performance of ReAct by itself being less impressive still means it is a useful contribution for the community.
∴ C: ReAct alone works fairly well for some tasks, but self-consistent Chain of Thought often outperforms ReAct alone, and leads to fairly dramatic increases in performance.

Proposed Method (w/ feedback):
P1: ReAct alone works fairly well for some tasks.
P2: Self-consistent Chain of Thought often outperforms ReAct alone.
P3: The best models use a smart combination of both CoT and ReAct.
P4: Models use heuristics like the 'ReAct → CoT–SC' strategy that maximize performance.
P5: If Self-consistent Chain of Thought often outperforms ReAct alone, and the best models use a smart combination of both CoT and ReAct, and models use heuristics like the 'ReAct → CoT–SC' strategy that maximize performance, then the performance of ReAct by itself is slightly less impressive.
P6: If ReAct alone works fairly well for some tasks, and the performance of ReAct by itself is slightly less impressive, and a technique can be a useful contribution even if its performance is less impressive, then ReAct is still a useful contribution for the community.
P7: A technique can be a useful contribution even if its performance is less impressive.
∴ C: The performance of ReAct by itself is slightly less impressive, and ReAct is still a useful contribution for the community.

Analysis

- For the direct reconstruction, both pairs of premises, P1 & P4, and P2 & P3, deductively imply an intermediate conclusion that "The performance of ReAct by itself is less impressive.". However, there is no logical connection between this intermediate conclusion and the final conclusion C, which indicates the reconstruction is *invalid* [×]. Furthermore, the final conclusion C does not correctly represent the original argument, which means the reconstruction is *unfaithful* [×].
- For the proposed method, the premises deductively imply the final conclusion (i.e., *valid* [✓]), and the reconstruction correctly yet completely represents the original argument (i.e., *faithful* [✓]), including the correct final conclusion.

Figure 6: Example #2 of automatic argument reconstruction.

Input: Argument (Review Point) & Paper

Paper:
Automatic Chain of Thought Prompting in Large Language Models (Zhang et al., 2023)

Argument (Review Point):
Finally, if clustering was the way to go, the authors could have tried to cluster samples into n-clusters, see where the test sample is fitting, and could have tried to use just one prompt instead of k different ones at each time. This might have reduced the compute cost and would be interesting to see the performance vs resource trade-off.

Output: Reconstructed Argument

Direct Reconstruction (w/o feedback):
P1: The authors could have tried to cluster samples into n-clusters and see where the test sample is fitting.
P2: The authors could have tried to use just one prompt instead of k different ones at each time.
P3: Using one prompt instead of k different ones would reduce the compute cost.
P4: If a method reduces compute cost, it would be interesting to see the performance vs resource trade-off.
P5: If the authors could have done something that would be interesting to see, then they should have done it.
$\therefore$ C: The authors should have tried to cluster samples into n-clusters, see where the test sample is fitting, and use just one prompt instead of k different ones at each time.

Proposed Method (w/ feedback):
P1: An alternative approach would be to cluster samples into n-clusters, see where the test sample is fitting, and use just one prompt instead of k different ones.
P2: If an alternative approach would use just one prompt instead of k different ones, then the alternative approach would reduce compute cost.
P3: If the alternative approach would reduce compute cost, then examining the performance vs. resource trade-off of the alternative approach would be interesting.
P4: If an alternative approach would use just one prompt instead of k different ones and examining the performance vs. resource trade-off of the alternative approach would be interesting, then the alternative approach would have been interesting to evaluate for its performance vs. resource trade-off.
$\therefore$ C: The alternative approach would have been interesting to evaluate for its performance vs. resource trade-off.

Analysis

- For the direct reconstruction, P3 and P4 imply an intermediate conclusion that "Using one prompt instead of k different ones would be interesting to see the performance vs resource trade-off.". Let us call this IC1. IC1 and P5 imply another intermediate conclusion that "The authors should have tried using one prompt instead of k different ones.". Let us call this IC2. Lastly, P1, P2, and IC2 imply the final conclusion C, indicating the reconstruction is *valid* [✓]. However, the final conclusion is stronger than what the original argument states. The original argument suggests an alternative approach, but the reconstructed final conclusion obligates authors to try it, meaning the reconstruction is *unfaithful* [×].
- For the proposed method, the reconstruction is *valid* [✓] and *faithful* [✓], including the correct degree of strength of the final conclusion.

Figure 7: Example #3 of automatic argument reconstruction.

D Dataset Details

D.1 Terms of Use and License

All papers and corresponding reviews used in our work are crawled from OpenReview[12], and our work is consistent with the Openreview terms of use[13]. Following these terms, we will release our work with a CC-BY 4.0 license.

D.2 Pilot Dataset

Before we construct the main dataset in Section 3.3, to ensure a trustworthiness and reliability of human annotation, we conduct a pilot study. We recruit three graduate students studying AI as human annotators, and let them choose total five papers submitted to ICLR 2021–2025[14]. Specifically, each human annotates reviews of three papers in OpenReview, where only one paper is authored by themselves and the other two non-authored papers are assigned in common. To ensure trustworthiness of non-authors' annotation, three humans annotate reviews and then we take a median value as a final human label.

D.3 Dataset Statistics

We present the REVIEWSCORE dataset statistics including the pilot and main subsets in Table 6. We include number of instances and percentage of *misinformed* labels.

Table 6: REVIEWSCORE dataset statistics.

	Pilot	Main	Total
ICLR Years	2021–2025	2021–2023	-
# Papers	5	40	45
# Review(er)s	19	155	174
Number of instances			
Review Points	84	573	657
Questions	22	121	143
Claims	5	87	92
Arguments	57	365	422
Premises	227	1,521	1,748

D.4 Full List of Papers

We provide a full list of papers used in the main subset of REVIEWSCORE dataset in Table 7. As explained in Section 3.3, eight papers are selected by each human annotator group with a common research interest.

[12] https://openreview.net
[13] https://openreview.net/legal/terms
[14] Unlike the main dataset, we include ICLR 2024–2025 since authors can access to their own submissions.

E Analysis of Human Annotators

E.1 Human Annotator Expertise

We report human annotator's expertise on REVIEWSCORE evaluation in Table 8. Specifically, for each annotator, we indicate a number of publications in AI/ML (including arXived works) and an averaged paper relevance score across eight assigned papers. We also indicate averaged numbers and scores for each group. The results show that an average number of publication is 3.93 and average paper relevance score is 4.06 / 5, indicating highly-experienced and relevant experts conduct a human annotation process. However, we observe that there are inter-group gaps in human expertise. Specifically, Group 2 and 4 shows significantly higher number of publications and paper relevance than other groups. Detailed score rubric for paper relevance is described in Figure 8, and we note that there is no instance where paper relevance of any human annotator is less than 3 (i.e., Moderate relevance).

Paper Relevance Score Rubric

5: Direct expertise – Works in the exact subtopic and can judge nuanced claims, methods, and datasets.
4: Strong relevance – Adjacent/overlapping subtopic with regular use of the paper's methods or domain; can evaluate technical choices with minimal ramp-up.
3: Moderate relevance – Same broad area (e.g., NLP ↔ NLP; CV ↔ CV) but different subtopic or methods; will understand contributions but may miss edge-case nuances.
2: Low relevance – Only tangential connection (e.g., general ML experience while the paper is domain-specific) and limited familiarity with core methods or domain.
1: No clear relevance – Outside the field; would require substantial background reading to assess claims/methodology.

Figure 8: Score rubric for evaluating paper relevance of human annotators.

E.2 Inter-Annotator Agreement

To ensure trustworthiness of human annotation, we report inter-annotator agreement in Krippendorff's Alpha (Hughes, 2021) on REVIEWSCORE evaluation in Table 9. Overall, REVIEWSCORE shows 0.489 Krippendorff's Alpha, indicating a moderate inter-annotator agreement. Specifically, QScore shows the highest agreement, ArgScore follows subsequently, and ClaimScore shows the lowest agreement.

By comparing inter-group agreements, Group 2 and 3 show significantly higher REVIEWSCORE

Table 7: Full list of ICLR submitted papers used in the REVIEWSCORE dataset.

Group	Topic	Paper Title
1	Image Generation	Lipman et al. (2023), Hertz et al. (2023), Couairon et al. (2023), Singer et al. (2023), Meng et al. (2022), Gal et al. (2023), Poole et al. (2023), Kim et al. (2023)
2	Time Series ML	Shukla and Marlin (2021), Zhang et al. (2022), Borisov et al. (2023), Nie et al. (2023), Nam et al. (2023), Zhang and Yan (2023), Wang et al. (2023), Shukla and Marlin (2022)
3	LLM Reasoning / Compression	Hsu et al. (2022), Ji et al. (2023), Lu et al. (2023), Ni et al. (2023), Fu et al. (2023), Liang et al. (2021), Ruis et al. (2023), Arora et al. (2023)
4	LLM / VLM	Hu et al. (2022), Kukleva et al. (2023), Wei et al. (2022), Sanh et al. (2022), Shen et al. (2022), Zhou et al. (2023b), Si et al. (2023), Shi et al. (2023)
5	LLM Prompting	Zhang et al. (2023), Yao et al. (2023), Zhou et al. (2023a), Razdaibiedina et al. (2023), Li et al. (2022), Jang et al. (2022), Creswell et al. (2023), Huang et al. (2023)

Table 8: Human annotator's expertise on REVIEWSCORE evaluation. A **bold** indicates the highest number/score across different groups, and an underline indicates the second highest.

Group	Annotator ID	# AI/ML Pub	Paper Relevance	Avg # Pub	Avg Paper Relevance
1	anno_11	2	3.50	3.00	4.00
	anno_12	3	4.13		
	anno_13	4	4.38		
2	anno_21	6	4.50	<u>5.33</u>	**4.38**
	anno_22	3	4.38		
	anno_23	7	4.25		
3	anno_31	0	4.00	1.33	3.96
	anno_32	1	4.00		
	anno_33	3	3.88		
4	anno_41	1	4.00	**6.67**	<u>4.04</u>
	anno_42	6	3.88		
	anno_43	13	4.25		
5	anno_51	3	3.13	3.33	3.92
	anno_52	1	4.13		
	anno_53	6	4.50		
Total	-	-	-	3.93	4.06

agreement than other groups. In contrast, Group 1 and 5 show lower agreement than other groups. Through a manual disagreement analysis, we confirm that most disagreements come from human annotators with low paper relevance (i.e., anno_11 and anno_51 in Table 8). This means that human annotations could be more reliable if their research interests become more relevant to the assigned papers. We leave this as a limitation of our work.

F Additional Results of Automatic ReviewScore Evaluation

F.1 Quantitative Results

We additionally report human-model agreement on REVIEWSCORE evaluation using different evaluation metrics in Table 10–13. Specifically, we report agreement on ClaimScore evaluation in Table 10, agreement on ArgScore evaluation in Table 11, agreement on QScore evaluation in Table 12, and agreement on REVIEWSCORE evaluation in Table 13. For the binary classification setup, we use Precision, Recall, and F1 Score, and for the 5-point scale setup, we use Pearson rank correlation, Gwet's AC2 (Gwet, 2001), and Quadratic Weighted Kappa (Warrens, 2012).

F.2 Qualitative Results

We present a thorough qualitative human-model disagreement analysis on REVIEWSCORE evaluation in Figure 9–14. For each review point type, we demonstrate two examples which include model-based scores and corresponding justifications using Claude Sonnet 3.7 and Claude Sonnet 4.

Table 9: Inter-annotator agreement (Krippendorff's Alpha) on REVIEWSCORE evaluation. A **bold** indicates the highest agreement across different groups, and an underline indicates the second highest.

Group	ClaimScore	ArgScore	QScore	REVIEWSCORE
1	0.392	0.339	0.357	0.357
2	0.438	**0.600**	**0.780**	**0.663**
3	**0.660**	0.457	0.655	0.580
4	0.356	0.535	0.554	0.489
5	0.235	0.306	0.335	0.254
Median	0.392	0.457	0.554	0.489

Table 10: Human-model agreement on ClaimScore evaluation.

	Binary			5-point Scale		
Model	Precision	Recall	F1	Pearson	AC2	Kappa
Proprietary models						
Claude Sonnet 3.7	0.091	**0.667**	0.160	0.217	0.056	0.156
Claude Sonnet 4	**0.167**	0.333	0.222	0.215	0.086	0.165
GPT-4o	0.000	0.000	0.000	0.098	0.036	0.093
GPT-5	0.091	0.200	0.125	0.020	-0.028	0.024
Gemini 2.5 Flash	0.158	0.500	**0.240**	**0.226**	**0.119**	**0.169**
Open-sourced models						
Qwen3-235B-A22B	**0.143**	**0.600**	**0.231**	0.191	0.096	0.142
Llama 3.3	0.083	0.200	0.118	0.083	0.063	0.097
DeepSeek-V3	0.000	0.000	0.000	**0.213**	**0.102**	**0.180**

Table 11: Human-model agreement on ArgScore evaluation.

	Binary			5-point Scale		
Model	Precision	Recall	F1	Pearson	AC2	Kappa
Proprietary models						
Claude Sonnet 3.7	0.357	0.652	0.462	0.323	0.238	0.261
Claude Sonnet 4	0.313	0.565	0.403	0.296	0.231	0.272
GPT-4o	**0.550**	0.239	0.333	0.230	0.237	0.244
GPT-5	0.419	0.565	**0.481**	0.401	0.342	0.353
Gemini 2.5 Flash	0.356	**0.674**	0.466	**0.469**	**0.387**	**0.402**
Open-sourced models						
Qwen3-235B-A22B	0.413	**0.413**	**0.413**	**0.295**	-0.016	0.148
Llama 3.3	**0.480**	0.261	0.338	-0.002	0.110	0.108
DeepSeek-V3	0.292	0.304	0.298	0.181	**0.155**	**0.176**

Table 12: Human-model agreement on QScore evaluation.

	Binary			5-point Scale		
Model	Precision	Recall	F1	Pearson	AC2	Kappa
Proprietary models						
Claude Sonnet 3.7	**0.576**	0.576	0.576	0.416	0.404	0.410
Claude Sonnet 4	0.524	0.647	**0.579**	**0.460**	**0.406**	**0.425**
GPT-4o	0.392	0.606	0.476	0.336	0.247	0.291
GPT-5	0.512	0.618	0.560	0.362	0.326	0.340
Gemini 2.5 Flash	0.393	**0.774**	0.522	0.330	0.201	0.265
Open-sourced models						
Qwen3-235B-A22B	0.400	**0.848**	**0.544**	0.331	0.132	0.243
Llama 3.3	0.380	0.794	0.514	**0.335**	**0.167**	**0.254**
DeepSeek-V3	**0.415**	0.733	0.530	0.277	0.067	0.192

Table 13: Human-model agreement on REVIEWSCORE evaluation.

Model	Binary			5-point Scale		
	Precision	Recall	F1	Pearson	AC2	Kappa
Proprietary models						
Claude Sonnet 3.7	0.367	0.622	0.462	0.368	0.315	0.336
Claude Sonnet 4	0.365	0.581	0.448	**0.414**	**0.349**	**0.378**
GPT-4o	0.392	0.378	0.385	0.351	0.340	0.347
GPT-5	**0.421**	0.565	**0.482**	0.362	0.327	0.341
Gemini 2.5 Flash	0.347	**0.699**	0.464	**0.415**	**0.349**	**0.378**
Open-sourced models						
Qwen3-235B-A22B	0.365	**0.595**	**0.452**	**0.351**	0.178	0.262
Llama 3.3	**0.370**	0.471	0.415	0.012	**0.299**	**0.311**
DeepSeek-V3	0.327	0.462	0.383	0.338	0.259	0.301

Input: Claim (Review Point) & Paper

Paper:
Towards Continual Knowledge Learning of Language Models (Jang et al., 2022)

Claim (Review Point):
The experimental findings in this paper are somewhat trivial.

Human-annotated Label (5-point Scale)

3, 4, 4 (Final: 4)

Output: ClaimScore Evaluation

Model: Claude Sonnet 3.7

Score: **2**

Human-model Disagreement Type: Value judgment

Justification: The proposition states that the experimental findings in this paper are somewhat trivial. To evaluate this claim, I need to assess the novelty, significance, and depth of the experimental findings presented in the paper.
Key experimental findings in the paper include:

...

These findings are not trivial because:
- They establish a new problem formulation and benchmark for an important real-world challenge
- They provide novel insights about how different methods perform in this setting
- They identify key factors affecting knowledge retention and acquisition
- The findings have practical implications for developing ever-changing language models

---

Model: Claude Sonnet 4

Score: **2**

Human-model Disagreement Type: Value judgment

Justification: To evaluate whether the experimental findings are "somewhat trivial", I need to assess the novelty, significance, and non-obviousness of the results presented in the paper.
Novel contributions and non-trivial findings:

...

Overall assessment:
While some high-level expectations (like catastrophic forgetting occurring) are predictable, the specific findings about which methods work best, the quantitative trade-offs, the multi-phase degradation effects, and the detailed analysis of what causes forgetting provide substantial non-trivial insights. The systematic benchmarking and the discovery that traditional CL methods don't transfer well to this domain are significant contributions.

The experimental work is comprehensive, the findings have practical implications for maintaining up-to-date language models, and several results challenge conventional wisdom from traditional continual learning.

Figure 9: Example #1 of ClaimScore evaluation.

Input: Claim (Review Point) & Paper

Paper:
Temperature Schedules for self-supervised contrastive methods on long-tail data (Kukleva et al., 2023)

Claim (Review Point):
The improvement is quite limited compared with other long-tailed learning methods.

Human-annotated Label (5-point Scale)

2, 2, 3 (Final: 2)

Output: ClaimScore Evaluation

Model: Claude Sonnet 3.7

Score: **3**

Justification: The proposition states that "The improvement is quite limited compared with other long-tailed learning methods". To evaluate this claim, I need to examine how the paper compares its proposed temperature scheduling (TS) approach with other long-tailed learning methods. Looking at the paper:

. . .

Given that the paper shows consistent improvements over the baselines and SDCLR, but doesn't compare with a broader range of long-tailed learning methods, and the improvements are modest (1-3%), the proposition that "the improvement is quite limited compared with other long-tailed learning methods" is partially supported but not fully substantiated.

Model: Claude Sonnet 4

Score: **4**

Human-model Disagreement Type: Value judgment

Justification: To evaluate this proposition, I need to examine the experimental results and compare the improvements achieved by the proposed temperature scheduling (TS) method against other long-tailed learning methods. Looking at the experimental results:

1. **Comparison with SDCLR**: In Table 3, the paper compares SimCLR+TS with SDCLR (a specialized method for long-tailed contrastive learning). The improvements are modest.
2. **Absolute improvements**: The improvements over baseline methods are generally in the range of 1–4% across different metrics and datasets.
3. **Limited baseline comparisons**: The paper primarily compares against standard contrastive learning methods (MoCo, SimCLR) and only one specialized long-tailed method (SDCLR). It doesn't compare against other established long-tailed learning approaches from supervised learning or other self-supervised long-tailed methods.
4. **Scope of evaluation**: The evaluation is limited to relatively small datasets (CIFAR variants and ImageNet-100) and doesn't demonstrate performance on larger-scale long-tailed datasets that are commonly used in long-tailed learning literature.
5. **Method simplicity vs. improvement**: While the method is simple (just scheduling temperature), the improvements are incremental rather than substantial breakthroughs.

The paper acknowledges this limitation implicitly by positioning the method as "simple yet effective" and emphasizing the zero additional computational cost, rather than claiming state-of-the-art performance improvements.

Figure 10: Example #2 of ClaimScore evaluation.

Input: Premise of Argument (Review Point) & Paper

Paper:
Automatic Chain of Thought Prompting in Large Language Models (Zhang et al., 2023)

Argument (Review Point):
Firstly, when mentioning that the method automatically finds prompts using chain of thoughts (Auto_CoT) approach, this does not imply finding the best samples from the dataset and using the same method as before. In other words, the proposed method does not find the best chain of thought prompts as claimed but rather believes that "let's do this step by step" is the best zero-shot-CoT prompting strategy and the method merely uses that with different prompt samples. The title and the abstract are highly misleading in that way.

Premise:
The method uses "let's do this step by step" with different prompt samples.

Human-annotated Label (5-point Scale)

5, 5, 5 (Final: 5)

Output: ClaimScore Evaluation

Model: Claude Sonnet 3.7

Score: **5**

Justification: The premise states that the Auto-CoT method uses "let's do this step by step" with different prompt samples. I need to check if this is accurate based on the paper. Looking at the paper, in the "DEMONSTRATION SAMPLING" section, the author describes how they construct demonstrations:

"A prompted input is formulated as: [Q: q(i)j. A: [P]], where [P] is a single prompt 'Let's think step to step'. This formed input is fed into an LLM using Zero-Shot-CoT (Kojima et al., 2022) to output the reasoning chain consisting of the rationale r(i)j and the extracted answer a(i)j."

. . .

So the premise is accurate - the Auto-CoT method does use "let's think step by step" (or a very similar variant "Let's think step to step") with different prompt samples selected through their clustering and sampling approach.

Model: Claude Sonnet 4

Score: **2**

Human-model Disagreement Type: Misunderstanding / incorrect reasoning

Justification: To judge the factuality of this premise, I need to examine what the Auto-CoT method actually does according to the paper. From the paper, I can see that:

1. Auto-CoT consists of two main stages: (i) question clustering and (ii) demonstration sampling
2. In the demonstration sampling stage, the method uses Zero-Shot-CoT to generate reasoning chains for selected questions
3. The paper explicitly states that Zero-Shot-CoT uses the prompt "Let's think step by step" (not "let's do this step by step")

. . .

So the premise is mostly accurate in describing the approach (using Zero-Shot-CoT with different samples), but contains a factual error in the exact wording of the prompt. The paper consistently uses "Let's think step by step" rather than "let's do this step by step."

Figure 11: Example #1 of ArgScore evaluation.

Input: Premise of Argument (Review Point) & Paper

Paper:
Automatic Chain of Thought Prompting in Large Language Models (Zhang et al., 2023)

Argument (Review Point):
Secondly, this is very obvious to anyone that the prompts must be representative of the test samples (as much as possible). This work takes the dataset, clusters them, finds the most representative sample from each cluster, and uses that as prompts. There is nothing extra-ordinary about this approach and is merely a small improvement over the randomly sampling strategy. This can also be seen from a marginal improvement in the results over a random sampling strategy.

Premise:
The approach is obvious (i.e., prompts must be representative of test samples).

Human-annotated Label (5-point Scale)

4, 4, 5 (Final: 4)

Output: ClaimScore Evaluation

Model: Claude Sonnet 3.7

Score: **2**

Human-model Disagreement Type: Misunderstanding / incorrect reasoning

Justification: I'll examine whether the paper itself acknowledges or implies that prompts should be representative of test samples as an obvious approach:

...

The paper's main contribution is showing that diversity in demonstrations is important, not just representativeness. The paper states: "diversity-based clustering may mitigate misleading by similarity" and "diversity of demonstration questions is the key."

This suggests that the premise oversimplifies the paper's findings. The paper doesn't claim that prompts must simply be representative – it argues for a balance of representativeness and diversity.

Model: Claude Sonnet 4

Score: **4**

Justification: To judge the factuality, I need to determine whether it's true that "prompts must be representative of test samples" is an obvious principle. This is a general statement about machine learning methodology and prompt engineering principles that would be known to AI practitioners before 2023. The idea that training/demonstration examples should be representative of test data is a fundamental principle in machine learning that predates this specific paper. This principle is well-established in the field and doesn't require knowledge of this specific paper to evaluate its truth.

The paper does implement this principle through clustering and selecting representative samples, but the general principle itself is established knowledge in the field.

Figure 12: Example #2 of ArgScore evaluation.

Input: Question (Review Point) & Paper

Paper:
ReAct: Synergizing Reasoning and Acting in Language Models (Yao et al., 2023)

Question (Review Point):
Is there a way to add self-consistency to ReAct, such that it samples multiple reasoning traces and actions and uses the majority action, for example? SC seems extremely effective for Chain of Thought tasks, and it seems like something similar could work here.

Human-annotated Label (5-point Scale)

2, 2, 5 (Final: 2)

Output: ClaimScore Evaluation

Model: Claude Sonnet 3.7

Score: **4**

Human-model Disagreement Type: Minority score prediction

Justification: The question asks about adding self-consistency to ReAct, specifically whether it's possible to sample multiple reasoning traces and actions and use the majority action, similar to how self-consistency works for Chain of Thought tasks. Let me check if the paper discusses this concept or something similar:

...

However, the paper does not explicitly discuss applying the self-consistency technique directly to ReAct itself (i.e., sampling multiple ReAct trajectories and taking the majority action). It only discusses combining ReAct with CoT-SC as separate methods, not applying the self-consistency sampling technique to ReAct's own trajectories.

Model: Claude Sonnet 4

Score: **4**

Human-model Disagreement Type: Minority score prediction

Justification: The question asks about adding self-consistency to ReAct, similar to how it's used in Chain of Thought (CoT) tasks. Let me examine what the paper says about self-consistency and whether it discusses or implements this for ReAct. Looking through the paper, I can see that:

...

However, the question specifically asks about applying self-consistency directly to ReAct itself – sampling multiple ReAct reasoning traces and actions and using majority voting. While the paper discusses CoT-SC and hybrid approaches that combine ReAct with CoT-SC, it doesn't appear to directly implement or discuss a "ReAct-SC" approach where multiple ReAct trajectories are sampled and majority voting is applied to the ReAct outputs themselves.

Figure 13: Example #1 of QScore evaluation.

Input: Question (Review Point) & Paper

Paper:
Large Language Models Can Self-improve (Huang et al., 2023)

Question (Review Point):
Can you check if baseline self-consistency with more examples can match LMSI or not?

Human-annotated Label (5-point Scale)

4, 5, 5 (Final: 5)

Output: ClaimScore Evaluation

Model: Claude Sonnet 3.7

Score: **2**

Human-model Disagreement Type: Misunderstanding / incorrect reasoning

Justification: The paper includes a hyperparameter study in Figure 4(b) that shows the effect of varying the number of sampled reasoning paths for self-consistency, both with and without LMSI. This directly addresses whether increasing the number of examples in baseline self-consistency can match LMSI performance.

Model: Claude Sonnet 4

Score: **4**

Justification: Looking at the experimental setup and results:
- The paper uses m=32 reasoning paths for generating training data
- Figure 4(b) shows accuracy vs number of sampled reasoning paths for both w/o LMSI and LMSI
- The experiments seem to use consistent numbers of paths for comparison

The paper doesn't systematically explore whether dramatically increasing the number of paths in baseline self-consistency (e.g., to 100+ paths) could match the LMSI performance. While Figure 4(b) shows diminishing returns after 15–20 paths, it doesn't explore very high numbers of paths that might be computationally equivalent to the LMSI training process.

Figure 14: Example #2 of QScore evaluation.

G Model Details

We measure reliability of automatic REVIEWSCORE evaluation with eight current state-of-the-art LLMs, including five proprietary models and three open-sourced models. For proprietary models, we use Anthropic's `claude-3-7-sonnet-20250219` (Anthropic, 2025a) and `claude-sonnet-4-20250514` (Anthropic, 2025b), OpenAI's `gpt-4o-2024-08-06` (OpenAI, 2024) and `gpt-5-2025-08-07` (OpenAI, 2025), and Google's Gemini 2.5 Flash (Gemini Team, Google, 2025). Since the REVIEWSCORE evaluation does not require longer chain of thought, we exclude reasoning models. Following this rule, we use Anthropic's claude models without thinking modes and OpenAI's gpt-5 model with minimal reasoning effort. For open-sourced models, we use Alibaba's `qwen3-235b-a22b-2507` (Yang et al., 2025) with 235B model parameters, DeepSeek's `deepseek-v3-0324` (DeepSeek-AI et al., 2024) with 671B model parameters and Meta's `llama-3.3-70b-instruct` (AI@Meta, 2024) with 70B model parameters. We call the models by their APIs.

H Prompts

We list every prompt used for automatic REVIEWSCORE evaluation (Section 4), automatic review point type classification (used in preprocessing stage of the dataset construction in Section 3.3), and automatic argument reconstruction (Section 3.2). Figure 15 – Figure 20 indicate prompts for automatic REVIEWSCORE evaluation and related score rubrics. Figure 21 and Figure 22 indicate prompts for automatic review point type classification, where the type is one of claim, argument, and question. Figure 23 – Figure 29 indicate prompts for automatic argument reconstruction. In the prompts, capital letters with double square brackets (i.e., [[XYZ]]) would be replaced by the corresponding material, and words with a curly bracket (i.e., {abc}) would be replaced by the corresponding score rubric (Figure 15 – Figure 17) or a python code snippet in Figure 26.

Unanswerability Score Rubric
1: Definitely Answerable by the paper
2: Probably Answerable by the paper
3: No Verdict
4: Probably Unanswerable by the paper
5: Definitely Unanswerable by the paper

Figure 15: Score rubric for evaluating *unswerability*.

Factuality Score Rubric
1: Definitely False
2: Probably False
3: No Verdict
4: Probably True
5: Definitely True

Figure 16: Score rubric for evaluating *factuality*.

Untrivialness Score Rubric
0: Definitely Trivial
1: Probably Trivial
2: Definitely Not Trivial

Figure 17: Score rubric for evaluating *untrivialness*.

Prompt for evaluating QScore (*unanswerability of questions*)

Paper
[[PAPER]]

Question
[[QUESTION]]

You are given a paper submitted to an AI conference and a question regarding the paper.

Judge if the question is answerable or not by the paper. You should scale 1-5 to indicate unanswerability as follows.
{Unanswerability Score Rubric}

If you score the question 1 or 2, then indicate which knowledge source you have grounded to (i.e., indicate corresponding section(s) and verbatim sentence(s)) and answer the question.

Your output should be the following.

Reasoning
[think step-by-step]

Unanswerability
[1 or 2 or 3 or 4 or 5]

Source
[corresponding section(s) and verbatim sentence(s) if the score is 1 or 2, otherwise None]

Answer
[answer the question if the score is 1 or 2, otherwise None]

Figure 18: Prompt used for evaluating QScore in a 5-point scale.

Prompt for evaluating ClaimScore or WScore (*factuality of weaknesses*)

Paper
[[PAPER]]

Weakness
[[WEAKNESS]]

You are given a paper submitted to an AI conference and a weakness regarding the paper.

Judge if the weakness is true or not based on the paper. You should scale 1-5 to indicate factuality as follows.
{Factuality Score Rubric}

Your output should be the following.

Reasoning
[think step-by-step]

Factuality
[1 or 2 or 3 or 4 or 5]

Figure 19: Prompt used for evaluating ClaimScore or WScore in a 5-point scale.

Prompt for evaluating ArgScore (*factuality of premises*)

Paper
[[PAPER]]

Weakness
[[WEAKNESS]]

Premise
[[PREMISE]]

You are given a paper submitted to an AI conference, a weakness of the paper, and one of premises of the weakness. Your task is to judge the factuality and untrivialness of the given premise.

First, judge the factuality of the premise. To do that, choose an appropriate knowledge source from:
1. given paper
2. annotator knowledge before the year [[YEAR]] (more precisely, before the paper is publicized)
3. other paper(s),

and then judge the factuality of the premise based on the knowledge source. You should scale 1-5 to indicate factuality as follows.
{Factuality Score Rubric}

Here are guidelines you should follow:
- Main purpose is to distinguish *given_paper* and *annotator_knowledge*.
- Select *other_papers* only if the premise refers a specific paper.
- Note that you should separate judging the factuality of the premise from understanding the semantics of the premise. It does not matter whether *given_paper* is needed or not to understand the semantics of the premise. The knowledge source is *given_paper* only if *given_paper* is needed to judge the factuality of the premise, otherwise, the knowledge source is *annotator_knowledge*. For example, although you need the paper's context in order to understand what the premise means, if you do not need the paper's knowledge to judge if the premise is true or not (e.g., logical assessment), then you should choose *annotator_knowledge* as a knowledge source and judge the factuality accordingly.
- For premises that are conditionals (If A then B), you should presume that the antecedent (A) is always true even if the antecedent does not align with the paper's knowledge. (Because the antecedent is always true, the knowledge source should only be determined while judging the factuality of the conseqeunt.) Then, choose an appropriate knowledge source to judge if the consequent (B) is true or not and judge the factuality accordingly.

Next, decide whether the premise is trivially true or not based on the common knowledge of CS/AI-majoring undergrad students before the year [[YEAR]] (more precisely, before the paper is publicized). For premises that are conditionals (If A then B), you must assume that the antecedent (A) is true and judge if the consequent (B) is trivially true or not. You should scale the score to 0-2 as follows.
{Untrivialness Score Rubric}

Here are guidelines you should follow:
- If the knowledge source is *given_paper* or *other_papers*, then untrivialness should always be 2 unless the premise factuality could also be determined by *annotator_knowledge*.
- If the knowledge source is *annotator_knowledge*, then untrivialness could be 0-2.

Your output should be formatted as below.

Reasoning
[think step-by-step]

Source
[given_paper or annotator_knowledge or other_papers]

Factuality
[1 or 2 or 3 or 4 or 5]

Untrivialness
[0 or 1 or 2]

Figure 20: Prompt used for evaluating ArgScore in a 5-point scale.

Prompt for classifying review point types (`is_argument`)

Review Point
[[REVIEW_POINT]]

You are given an AI conference review point. Is this an argument or not? Your response should follow the format below.

Reasoning
[think step-by-step]

Response
[Yes or No]

Figure 21: Prompt used for deciding if a review point is an argument or not (`is_argument`).

Prompt for classifying review point types (`is_question`)

Review Point
[[REVIEW_POINT]]

You are given an AI conference review point. Decide if this is a question or a simple claim. Your response should follow the format below.

Reasoning
[think step-by-step]

Response
[Question or Claim]

Figure 22: Prompt used for deciding if a review point is a question or a claim (`is_question`).

Prompt for Argument Reconstruction (`extract_verbatim_conclusion_reason`)

Review Point
[[REVIEW_POINT]]

Given an AI conference review point, consider it as an argument, and then returns its verbatim conjecture in the source text and verbatim reason statements of that conjecture in the source text. The output format should be as following.

Conjecture
[main conjecture in the review point]

Supporting Reasons
[list of supporting reasons for the conjecture]

Figure 23: Prompt used for extracting verbatim conjecture and reason statements in an argument (`extract_verbatim_conclusion_reason`).

Prompt for Argument Reconstruction (`argument_reconstruction`)

Paper
[[PAPER]]

Review Point

Conclusion
[[CONCLUSION]]

Explicit reasons
[[REASONS]]

You are given a paper submitted to an AI conference and a review point by a peer reviewer. A review point consists of a conclusion and its explicit reasons. Reconstruct an argument (i.e., a review) with premise-conclusion structure where premises deductively imply the conclusion. The reconstructed argument should be deductively valid, using formal logical patterns like modus ponens (e.g., Premise1: A, Premise2: If A then B, Conclusion: B). Add implicit premises and intermediate conclusions if needed.

Your output should composed of two parts, argument reconstruction and its formalization. In the first part, list premises, intermediate conclusions, and conclusion, and indicate their logical connection (i.e., which propositions deductively implies which). In the second part, first define variables and/or predicates, then formalize premises, intermediate conclusions, and a conclusion, and then generate a deductive proof. The output format should be as following.

Argument Reconstruction

Premises
[list of explicit and implicit premises]

Intermediate Conclusions
[list of intermediate conclusions (if intermediate conclusions are not needed, then write "None".)]

Conclusion
[a conclusion]

Logical Connections
[list of logical connections]

Formalized Argument

Defined Variables/Predicates
[definition of each variable and/or predicate]

Formalized Premises
[formalization of premises using definition]

Formalized Intermediate Conclusions
[formalization of intermediate conclusions using definition (if intermediate conclusions are not needed, then write "None".)]

Formalized Conclusion
[formalization of conclusion using definition]

Deductive Proof
[deductive proof using formalized premises]

Figure 24: Prompt used for reconstructing an argument (`argument_reconstruction`).

Prompt for Argument Reconstruction (`streamlining`)

Defined Variables/Predicates
[[DEFINITION]]

Formalized Premises
[[PREMISES]]

Formalized Conclusion
[[CONCLUSION]]

Deductive Proof
[[PROOF]]

First, determine necessary formalized premises for the given deductive proof. This includes:
1. Add any missing formalized premises that are necessary to prove conclusion but cannot be dervied from the formalized premises.
2. Remove any unnecessary formalized premises that are not necessary to prove conclusion but present in the formalized premises.

You should format these premises into a python dictionary where keys and values are python strings.

Second, write a python program using z3 that inputs the necessary formalized premises and formalized conclusion and outputs:
1. Their validity, formatted as a python string of either "valid" or "invalid".
2. A smallest subset of necessary formalized premises to prove the formalized conclusion, formatted as a python list of keys of the python dictionary of the necessary formalized premises.

You should therefore print two things (a python string and a python list) separately. Please use the below python code snippet.
{Code snippet for checking validity}

Third, return the final formalized conclusion that is used in the python program in step 2.

Lastly, judge whether the formal proof using the necessary formalized premises (in step 1) and the final formalized conclusion (in step 3) is circular or not. If there is a single necessary formalized premise that is the same as the final formalized conclusion, then return N/A.

Your response should be as following.

Necessary Formalized Premises
```` python
{
    "[Symbol of a premise #1]": "[Formalization of a premise #1]",
    "[Symbol of a premise #2]": "[Formalization of a premise #2]",
    ...
}
```

### Python Program
```` python
[a python program]
```

### Final Formalized Conclusion
[Formalized conclusion in the python program]

### Proof Circularity
[Yes or No or N/A]

Figure 25: Prompt used for streamlining formalized reconstruction (`streamlining`).

**Code snippet for checking validity**

```
```` python
from z3 import *
import itertools

##################################
### Write down your code here ###
##################################

# Check validity of the argument
def check_validity(premises_dict, conclusion):
    s = Solver()
    s.add(list(premises_dict.values()))
    s.add(Not(conclusion))
    if s.check() == unsat:
        return "valid"
    else:
        return "invalid"

# Find minimal set of premises
def find_minimal_premises(premises_dict, conclusion):
    for subset_size in range(1, len(premises_dict) + 1):
        for subset in itertools.combinations(premises_dict.keys(), subset_size):
            subset_premises = [premises_dict[key] for key in subset]
            if check_validity(subset_premises, conclusion) == "valid":
                return list(subset)
    return list(premises_dict.keys())

validity = check_validity(premises, conclusion)
print(validity)
minimal_premises = find_minimal_premises(premises, conclusion)
print(minimal_premises)
```
```

Figure 26: Python code snippet for evaluating validity of reconstruction.

**Prompt for Argument Reconstruction (`program_debugging`)**

```
When I execute the python program, I got the following error:
[[ERROR]]

Fix the error and generate a revised python program. Your response should be as following.

### Reasoning
[explain why and how to fix the program]

### Revised Python Program
```` python
[a python program]
```
```

Figure 27: Prompt used for debugging python programs that evaluate validity of reconstruction (`program_debugging`).

**Prompt for Argument Reconstruction (`deformalization`)**

## Defined Variables/Predicates
[[DEFINITION]]

## Formalized Premises
[[PREMISES]]

## Formalized Conclusion
[[CONCLUSION]]

Given definitions of variables and/or predicates, generate natural language (NL) descriptions of formalized premises and conclusion. Your response should be as following.

### NL Premises
[list of premises in natural language]

### NL Conclusion
[conclusion in natural language]

Figure 28: Prompt used for translating FOL formulas with keys (i.e., defined variables/predicates) to NL reconstructed arguments (`deformalization`).

**Prompt for Argument Reconstruction (`check_faithfulness`)**

# Argument
[[ARGUMENT]]

# Argument Reconstruction

## Premises
[[PREMISES]]

## Conclusion
[[CONCLUSION]]

For an argument, its reconstruction as a premise-conclusion structure is given. Your task is to judge whether the construction is faithful or not. You should judge the faithfulness according to the following two criteria:
- **Accuracy & Charity.** The reconstruction should keep the author's intended meaning while eliminating irrelevancies—i.e., obey the principle of charity and prefer the strongest sensible reading of ambiguous passages.
- **Completeness.** All explicit premises, the main conclusion and any indispensable implicit premises must be included.

The output format should be as following.

# Reasoning
[Explain step-by-step]

# Faithfulness
[Yes or No]

Figure 29: Prompt used for evaluating faithfulness of reconstruction (`check_faithfulness`).