Evaluating AI Agents with LLM-Based Judges

Prashant Kafle
Department of Computer Engineering
Thapathali Campus, IOE
Kathmandu, Nepal
Roll no: THA080BCT026
prashantkafle7738@gmail.com

Sunil Paudel
Department of Computer Engineering
Thapathali Campus, IOE
Kathmandu, Nepal
Roll no: THA080BCT047
sunil.080bct047@tcioe.edu.np

***Abstract*—As language-model agents take on longer tasks, evaluating their outputs becomes a practical systems problem. Human assessment is expensive and difficult to repeat at scale, while automatic metrics can miss factual errors, useful reasoning, and task-specific quality. This paper reports on two linked but methodologically separate studies conducted on a shared benchmark of agent tasks sampled from GAIA [13]. Study 1 is a purely algorithmic comparison: ROUGE-L measures reference-based similarity with longest common subsequence dynamic programming, Bradley–Terry and Elo-style models convert pairwise judgments into rankings, and SelfCheckGPT (here, a lexical token-overlap proxy rather than the original NLI-based scorer) uses disagreement among stochastic samples to detect possible hallucinations, applied to 18 agent responses generated from 18 stratified GAIA tasks by a single agent configuration, together with a $k = 3$ resampling of an 8-task subset used for the SelfCheckGPT and per-task Bradley–Terry analysis. Study 2 is a judge–human agreement study: two trained human annotators (the paper's authors) and a Groq-based LLM judge independently label a 30-item primary pool drawn from those responses and resamples under a fixed four-category rubric, reported with Cohen's kappa, per-class precision/recall/F1, and a binary-vs-categorical agreement breakdown; a secondary, quota-limited 10-item subsample additionally includes a Gemini judge for an inter-judge comparison. Because two author-annotators cannot serve as an independent, unbiased gold standard, and because both studies use one agent configuration on one benchmark, results are reported as evidence about this specific setup rather than general claims about LLM judges. The two studies are designed, sampled, and analyzed independently, so an algorithmic result from Study 1 is never mistaken for evidence of judge validity in Study 2.**

***Index Terms*—agent evaluation, LLM-as-a-judge, ROUGE-L, Bradley–Terry model, SelfCheckGPT, Cohen's kappa**

I. Introduction

Modern agents can plan, call tools, and revise their own outputs. That makes evaluation harder than checking whether a short answer matches a reference. An agent may produce a correct final answer with different wording, or a plausible answer that contains one unsupported claim. It may also succeed on one run and fail on another because sampling, tool responses, or intermediate decisions change. A useful evaluation method therefore needs to measure more than surface similarity.

Human review remains the most direct way to judge whether an agent followed instructions and produced a useful result. It is slow, expensive, and difficult to apply consistently across many runs. This has encouraged the use of automatic metrics and, more recently, large language models as evaluators. An LLM judge can read an open-ended response and return a score, a comparison, or a written explanation. The same flexibility that makes this approach attractive also creates a problem: the judge may favor certain styles, follow superficial cues, or agree with another model without tracking factual correctness.

This paper treats evaluation as a sequence of related algorithmic problems. ROUGE-L provides a low-cost reference-based score using the longest common subsequence (LCS). Pairwise model comparisons can be aggregated with Bradley–Terry or Elo-style ranking models. SelfCheckGPT tests whether sampled answers stay consistent when there is no external fact database to check against. The last piece, Cohen's kappa, is arguably the one that matters most: it measures whether the automated judge's decisions actually agree with human annotations. A judge that produces stable scores but disagrees with people is not reliable, no matter how consistent it looks on its own.

The study implements these methods and compares their computational behavior and practical limitations on agent-generated responses. The main research question is: *under what conditions do automatic and LLM-based evaluation methods provide measurements that are both scalable and aligned with human judgment?* The analysis has four goals:

- implement ROUGE-L and report its dynamic-programming cost and sensitivity to wording;
- estimate model rankings from pairwise judgments and examine how rankings change with the comparison data;
- test whether SelfCheckGPT's consistency signal identifies hallucinated or unsupported claims; and
- measure judge–human agreement with Cohen's kappa instead of treating an LLM score as ground truth.

The remainder of the paper reviews prior work, describes the algorithms and experimental design, reports results, and discusses the limits of automated evaluation.

II. Literature Review

A. Two Layers of Evaluation

Prior work on evaluating language-model agents can be separated into two layers that are frequently discussed together

but ask different questions. The first layer evaluates *agent outputs*: does a response match a reference, does it rank above a competing response, and is it internally consistent enough to be trusted as a factual claim? The second layer evaluates *the evaluator itself*: when an LLM is used to produce those judgments, is the judge's own output reliable, and does it agree with independent human decisions? Blending these two layers makes it easy to mistake a stable or fluent judgment for a correct one. This paper's four-part pipeline follows the same split: ROUGE-L, the Bradley–Terry/Elo ranking models, and SelfCheckGPT belong to the first layer, while LLM-as-a-judge reliability and Cohen's kappa belong to the second. The rest of this section reviews prior work under each layer in turn.

B. Layer 1: Evaluating Agent Outputs

1) Reference-Based Text Evaluation: ROUGE was introduced as a package for comparing generated summaries with human-written references [1]. ROUGE-L uses the longest common subsequence between a candidate and a reference. Unlike exact n-gram overlap, the LCS can reward a shared sequence even when words are separated by insertions. Its dynamic-programming computation is straightforward: for sequences of lengths m and n, the standard LCS table requires $O(mn)$ time and $O(mn)$ space, or $O(\min(m, n))$ space with row reuse. These properties make ROUGE-L useful as a reproducible baseline.

Its limitation is equally important for agent evaluation. ROUGE-L measures overlap with a reference, not whether a response is factually correct, executable, or useful. Two valid answers can receive different scores because they use different phrasing. A response can also share many words with a reference while containing a subtle error. Recent surveys therefore place lexical metrics among the efficient tools for constrained tasks, while noting their weakness on open-ended generation [6].

2) Pairwise Comparison and Ranking: When a single numeric reference score is unavailable, evaluators can compare two outputs directly. The Bradley–Terry model represents the probability that item i defeats item j as

$$P(i \succ j) = \frac{\exp(\beta_i)}{\exp(\beta_i) + \exp(\beta_j)}, \tag{1}$$

where β_i and β_j are latent quality parameters [2]. Fitting these parameters over many comparisons produces a ranking and exposes uncertainty caused by sparse or uneven comparisons. Elo ratings use a related update rule and are widely associated with competitive pairwise ranking [3].

Pairwise evaluation is attractive because people often find it easier to choose the better of two answers than to assign an absolute score. This idea appears in large-scale language-model evaluation, including MT-Bench and Chatbot Arena, where model comparisons are aggregated into rankings [4]. However, rankings inherit the judge's biases. Position effects, verbosity preferences, and inconsistent criteria can change the result. A ranking algorithm can summarize judgments accurately while the judgments themselves remain poorly grounded, a concern that belongs to the second layer of evaluation discussed below.

3) Consistency-Based Hallucination Detection: SelfCheckGPT addresses hallucination detection without requiring model probabilities or an external knowledge base [5]. The method samples multiple responses to the same prompt and compares the claims across samples. If a model has stable knowledge of a fact, the samples should tend to agree. Unsupported claims are more likely to vary or contradict one another. The approach can therefore be applied to black-box systems for which token probabilities are unavailable.

The method does not establish truth by itself. A model can repeat the same false statement in every sample, and legitimate answers may vary in wording. Sampling also adds computation: producing k responses costs roughly k model calls before the consistency comparisons are performed. These tradeoffs make SelfCheckGPT a useful diagnostic signal rather than a replacement for fact verification. This study examines both its detection quality and its cost as the number of samples changes.

C. Layer 2: Evaluating the Evaluators

1) LLM-as-a-Judge and Its Own Reliability: LLM-as-a-judge systems use a language model to score, rank, or select outputs, and increasingly stand in for the ranking and consistency judgments described in Layer 1. Surveys of the area describe applications in open-ended generation and identify reliability, bias, prompt design, and judge calibration as unresolved concerns [7], [8]. The central distinction is between using an LLM to generate a judgment and showing that the judgment is valid. A fluent explanation or a stable score does not prove agreement with a human evaluator, which is precisely why the judge itself, and not only the outputs it scores, must be treated as an object of study.

2) Benchmarks That Test Judges Directly: A growing line of work targets the judge as the unit of evaluation rather than the agent under test. JudgeBench constructs challenging response pairs with objectively correct preference labels and shows that strong LLM judges, including GPT-4o, perform only slightly better than random guessing on cases where crowdsourced preference is a poor proxy for factual and logical correctness [10]. Agent-as-a-Judge extends this idea to agentic, multi-step tasks: it uses an agentic evaluator that inspects intermediate steps rather than only the final output, and reports that this step-aware judge is more reliable than a conventional LLM-as-a-judge baseline and comparable to human evaluation on the DevAI benchmark [11]. AgentRewardBench evaluates LLM judges specifically on web-agent trajectories against expert annotations and finds that no single judge performs well across all benchmarks, and that the rule-based evaluators commonly used as a substitute tend to underreport agent success [12]. Together, these benchmarks make the same point from three different task settings: a judge's fluency or apparent consistency is not evidence of its correctness, and judges require dedicated benchmarks in the same way that agents do.

3) Chance-Corrected Human Agreement: Cohen's kappa provides a lightweight way to test judge–human agreement for

categorical labels while correcting for agreement expected by chance [9]. For observed agreement p_o and chance agreement p_e, it is defined as

$$\kappa = \frac{p_o - p_e}{1 - p_e}. \tag{2}$$

Unlike raw accuracy, kappa can reveal when a judge appears accurate mainly because one label dominates the dataset. Its interpretation still depends on the label distribution and annotation protocol, so it should be reported with the confusion matrix and class counts. Compared with dedicated judge benchmarks such as JudgeBench and AgentRewardBench, kappa is a much smaller-scale check, but it plays the same role: it is the validation measure that connects the Layer 1 algorithms to the practical question of trust, namely whether an automated judge agrees with human decisions on the same outputs.

D. Research Gap

Existing work tends to stay within one of the two layers, and this separation is the concrete gap this paper targets. Reference metrics, pairwise ranking models, and hallucination detectors study the outputs of an agent; judge benchmarks such as JudgeBench, Agent-as-a-Judge, and AgentRewardBench study the reliability of the judge in isolation, typically on a single task family such as knowledge-and-reasoning pairs, coding, or web-agent trajectories [10]–[12]. None of these studies test whether the same LLM judge that produces Layer 1 pairwise labels (feeding the Bradley–Terry ranking) or Layer 1 consistency judgments is also validated, on the same task set, against independent human labels. This is a specific and currently unanswered empirical question, not a rhetorical one: a judge could produce Bradley–Terry rankings that remain stable across bootstrap resamples of the comparison data while still disagreeing with human annotators beyond what chance predicts, and no published study we found reports both quantities on the same benchmark. The four methods studied here are therefore not grouped by convenience; they are grouped because Study 2 (Section IV-D) directly tests the judge that Study 1 (Section IV-C) partially depends on to produce its pairwise and consistency labels. The two studies are reported separately, on the same GAIA-derived benchmark, precisely so that a favorable result in one is never substituted for evidence in the other.

III. Theoretical Framework and Justification

The four methods studied in this paper are not arbitrary choices. Each targets a distinct layer of the evaluation problem, and together they form a chain that moves from surface-level text comparison to a validated measure of judge trustworthiness. This section states the theoretical basis for that chain and justifies why each algorithm is necessary rather than redundant.

A. Evaluation as a Measurement Problem

Treating agent evaluation as a measurement problem, rather than a scoring exercise, motivates the overall design. A measurement instrument must satisfy two properties: it must be computable at reasonable cost, and its output must correspond to the quantity it claims to measure. Classical test theory separates these concerns into reliability, whether repeated measurement gives consistent results, and validity, whether the measurement reflects the target construct [9]. ROUGE-L and Bradley–Terry/Elo scores are efficient and reliable by construction, since they are deterministic functions of fixed inputs. Their validity, that is, whether a high score reflects real response quality, is not guaranteed and must be checked separately. SelfCheckGPT targets a different construct, factual consistency, that similarity and ranking scores do not capture. Cohen's kappa is included specifically to test validity: it measures whether the judge's categorical decisions align with human decisions beyond chance, which reliability alone cannot establish. This framework explains why no single metric in the pipeline is sufficient on its own.

B. Complexity and Feasibility Justification

Each algorithm is chosen partly because its computational profile matches what is feasible to run repeatedly over an agent's outputs. The LCS dynamic program behind ROUGE-L runs in $O(mn)$ time, which is negligible next to the cost of generating the responses themselves, so it is used as a cheap first-pass filter. Bradley–Terry and Elo estimation are used because pairwise comparison data can be collected incrementally, and maximum-likelihood fitting of the Bradley–Terry model or iterative Elo updates scale close to linearly in the number of comparisons, which keeps ranking updates tractable as more agent outputs are produced. SelfCheckGPT is more expensive, since it requires k additional model samples per prompt, so it is justified only for the subset of outputs where a similarity or ranking score is inconclusive, not for every response. Cohen's kappa is computed once over a fixed confusion matrix and adds negligible cost, which is why it is used as the final validation step rather than an early filter. Ordering the pipeline from cheapest to most expensive is itself a design decision grounded in algorithmic complexity, not only in what each method measures.

C. Why Pairwise Comparison Over Absolute Scoring

The choice of Bradley–Terry and Elo models over direct absolute scoring follows from known limitations of asking a judge, human or LLM, to assign a number on a fixed scale. Absolute scores from different judges or different sessions are not guaranteed to be on a common scale, and a single judge's scale can drift across a long batch of items. Pairwise comparison avoids anchoring an output to an arbitrary numeric scale; it only requires a local decision about which of two outputs is better, and the ranking is recovered afterward through the probabilistic model in Equation (1). This is the same justification used in large-scale human preference benchmarking [4], and it applies equally when the judge is an LLM rather than a person.

D. Why Consistency Substitutes for Ground Truth

SelfCheckGPT is justified by a specific theoretical assumption: when a language model has genuinely learned a fact,

resampling under different random seeds should reproduce that fact, while an unsupported or fabricated claim is less likely to be reproduced identically across samples [5]. This assumption lets the method detect a likely hallucination without an external database, at the cost of being a proxy rather than a direct correctness check. The framework used here treats SelfCheckGPT's output as an additional feature alongside ROUGE-L and ranking scores, not as a standalone verdict, which is consistent with its original design as a sampling-based diagnostic rather than a certifier of truth.

E. Why Human Agreement Is the Final Criterion

The theoretical framework treats the LLM judge as a hypothesis to be tested, not as an oracle. This follows directly from inter-rater reliability theory, where any single rater's output is only meaningful once compared against another rater using a chance-corrected statistic [9]. Applying Cohen's kappa between the LLM judge and human annotators operationalizes this idea: a kappa near zero indicates that apparent judge accuracy is an artifact of label imbalance, while a high kappa supports treating the judge's categorical decisions as a usable substitute for costly human review in future evaluations. This is why the pipeline ends with kappa rather than with the judge's own score.

F. Why Two Studies Rather Than One Pipeline

The four methods belong together theoretically, but they do not belong in a single empirical design. ROUGE-L, Bradley–Terry/Elo, and SelfCheckGPT are deterministic or sampling-based algorithms applied to fixed inputs; evaluating them requires no human subjects and answers questions about computational cost and statistical behavior. Cohen's kappa evaluates a different kind of claim, namely whether a specific judge's categorical decisions agree with independent human decisions, and answering it requires a human-subjects annotation protocol with its own sampling, blinding, and reliability checks. Reporting these under one undifferentiated "Research Design" section, as an earlier draft of this paper did, made the four methods read as unrelated case studies rather than as a chain. Section IV-C therefore reports Study 1 (the algorithmic comparison) on its own data subsample and its own statistical tests, and Section IV-D reports Study 2 (the judge-agreement study) on its own subsample, annotation protocol, and statistical tests. The two studies are connected only through the research question stated in Section I and through the fact that the LLM judge validated in Study 2 is the same judge used to generate the pairwise labels analyzed in Study 1.

IV. Research Design

A. Overview

The study is organized as two separate studies that share a single source of agent tasks (Section IV-B) but differ in design, sampling, and analysis. Study 1 (Section IV-C) is a purely algorithmic comparison of ROUGE-L, Bradley–Terry/Elo ranking, and SelfCheckGPT; it requires no human subjects and answers questions about computational cost and statistical behavior. Study 2 (Section IV-D) is an LLM judge–human agreement study; it requires an independent human-annotation protocol and answers whether the LLM judge agrees with human decisions beyond chance. Results, procedures, and statistical tests for the two studies are reported in separate subsections so that they are never conflated.

B. Shared Task Source and Inclusion Criteria

Both studies draw their agent tasks from the validation split of the GAIA benchmark [13], which provides 165 questions with public, human-verified reference answers across three difficulty levels (Level 1: single-step lookup; Level 2: short multi-step reasoning or tool use; Level 3: long multi-step tool chains). GAIA is used instead of a self-constructed task set because its reference answers and difficulty stratification are already documented and independently verifiable, which supports the reproducibility of task selection.

A GAIA task is included in the study if it satisfies all of the following criteria:

- it is text-only, i.e. answerable using only the question text and web search, with no attached image, audio, or document file (the evaluation sandbox provides a web-search tool, not general file processing);
- its published reference answer is a short, checkable string suitable for exact-match or ROUGE-L scoring;
- the agent under evaluation returns a non-empty final answer using a bounded budget of web-search calls and a single LLM call.

Applying these criteria to the GAIA validation split yielded a working set of 18 tasks, stratified across the three difficulty levels (6 Level 1, 8 Level 2, and 4 Level 3 tasks). Each included task is run through a single agent configuration: one LLM call (Groq-hosted Llama 3.3 70B) augmented with DuckDuckGo web-search snippets. This produces the 18 agent responses that form the base dataset for Study 1. The original proposal compared two agent configurations (a baseline and a tool-augmented variant); this was cut to one configuration to keep the study feasible within a six-day scope, so Study 1 reports how the four evaluation methods behave on a single agent rather than a cross-configuration comparison. A further subset of 8 tasks, drawn from the same 18, is resampled $k = 3$ times to support the SelfCheckGPT consistency check and a per-task Bradley–Terry ranking of the three resamples.

C. Study 1: Algorithmic Comparison of Output-Evaluation Methods

1) Design: Study 1 answers a purely algorithmic question: how do ROUGE-L, Bradley–Terry/Elo ranking, and SelfCheckGPT behave, in terms of computational cost, score distribution, and ranking stability, on the same set of agent outputs? Study 1 uses no human labels and computes no Cohen's kappa; the pairwise labels it produces with the LLM judge are simply the object that Study 2 later validates.

2) Procedure:

- **Stage 1 – Reference scoring.** ROUGE-L is computed between each of the 18 agent responses and its GAIA reference answer using the LCS-based dynamic program. The full score distribution (mean, median, count of non-zero scores) is recorded.
- **Stage 2 – Pairwise ranking.** Because only one agent configuration is evaluated (Section IV-B), Bradley–Terry ranking is applied per task instead of across configurations: for each task in the 8-task SelfCheckGPT subset, the LLM judge compares the $k = 3$ resampled responses pairwise, and a within-task Bradley–Terry strength is fit from the comparisons that survive a position-bias control (a comparison is kept only if the judge's verdict is unchanged when the response order is swapped).
- **Stage 3 – Consistency checking.** For the same 8-task, $k = 3$ resampled subset, SelfCheckGPT compares claims across the three samples per response and produces a consistency score per response. The implementation used here is a lexical Jaccard token-overlap proxy rather than the original NLI-based scorer, so the consistency score reflects surface word overlap between samples rather than the semantic entailment check of [5].

3) Statistical Analysis Plan: ROUGE-L scores are summarized descriptively (mean, median, count of non-zero scores) over the 18 tasks. Because only one agent configuration is evaluated, no paired cross-configuration test applies; the originally planned 500-resample bootstrap Kendall's τ over cross-configuration rankings is dropped for the same reason and reported here as a scope reduction rather than a result that was not run. Per-task Bradley–Terry fits are instead reported as a yield statistic: the fraction of the 8-task subset for which at least one position-bias-consistent comparison was obtained. SelfCheckGPT consistency scores are summarized descriptively and, for the same 8-task subset, compared qualitatively against the corresponding human hallucination labels from Study 2 as a small, non-circular sanity check; at $n = 8$ the subset is too small to support a formal correlation test.

D. Study 2: LLM Judge–Human Agreement Study

1) Design: Study 2 answers a different question from Study 1: does the LLM judge's categorical decision on a response agree with an independent human decision on the same response, beyond the agreement expected by chance? Study 2 is reported with its own subsample, its own annotators, and its own statistical test (Cohen's kappa), so that a favorable algorithmic result in Study 1 is never substituted for evidence of judge validity.

2) Sampling: A pool of 42 candidate items is built by combining the 18 main Study 1 responses with the 24 SelfCheckGPT resamples (8 tasks $\times$ $k = 3$) from Section IV-C, since the 18 main responses alone are fewer than the roadmap's 30-item annotation target. From this pool, the first 6 items (post-shuffle) are set aside for calibration, and a further 30 of the remaining 36 candidates form the primary annotation sample: both human annotators and the Groq-based LLM judge label all 30. The other 6 candidates are not used. Gemini's free-tier quota (empirically confirmed to be 20 requests/day for this project) makes labeling the full 30-item sample impractical, so a second, smaller comparison uses only the 10 main-response items, a subset of the 30, to add a Gemini judge and a Groq–Gemini inter-judge check as a secondary, quota-limited robustness analysis. The 30-item and 10-item results are reported separately (Section V-B) and are never pooled, since they differ in size and item composition.

3) Rubric: Each response is labeled with exactly one of four categories, applied identically by both human annotators and the LLM judge: (1) *Correct* – matches the reference or task intent with no unsupported claims; (2) *Partially correct* – the main answer is right but includes a minor unsupported or extraneous claim; (3) *Incorrect* – the main answer does not match the reference or fails the task; (4) *Hallucinated* – the response contains a specific factual claim not supported by the reference, the retrieved tool output, or the task context.

4) Annotation Protocol: The two human annotators are the paper's authors. Two author-annotators, even with the controls below, do not constitute an independent, unbiased gold standard; the protocol reduces but cannot eliminate the risk of shared bias:

- **Calibration.** Both annotators jointly label a separate calibration set of 6 responses (disjoint from the 30-item primary sample), discuss disagreements, and refine the rubric wording before the main annotation round.
- **Independent, blind labeling.** For the 30-item primary sample, each annotator labels all responses independently, in an independently randomized order, without access to the LLM judge's labels or to the other annotator's labels.
- **Human–human reliability precondition.** Cohen's kappa is computed between the two human annotators before any comparison with the LLM judge. If human–human kappa falls below 0.6 (the conventional threshold for "substantial agreement" on the Landis–Koch scale), the rubric is judged too unreliable to serve as a reference standard; it is revised and re-piloted on a fresh calibration set before Study 2 proceeds to the judge comparison.
- **Adjudication.** Where the two annotators disagree, an adjudicated label is produced through discussion, guided by a fixed decision rule: *correct* requires the response to match the GAIA reference answer; *hallucinated* requires a specific claim in the response that is absent from both the reference answer and the search-tool context returned to the agent; *incorrect* covers any other mismatch with the reference answer that introduces no such specific unsupported claim. This adjudicated label, not either individual annotator's label, is the human reference standard used against the LLM judge. No per-item written rationale beyond the adjudicated label itself was logged during adjudication, which we note as a protocol limitation in Section VI rather than presenting the adjudicated labels as independently auditable.

The Groq-based LLM judge labels the same 30 items using the same rubric text, without access to either annotator's labels, in

a single pass. The Gemini-based LLM judge labels only the 10-item main-response subset, for the secondary comparison described in Section V-B.

5) Statistical Analysis Plan: Two kappa statistics are reported as point estimates over the full 30-item primary sample, without a bootstrap confidence interval given the sample size:

- *Human–human kappa*, comparing the two annotators before adjudication over all 30 items, reported as evidence that the rubric is applied reliably;
- *Groq judge–human kappa*, comparing the Groq judge's labels with the adjudicated human labels over all 30 items, reported as the paper's central validity result.

Both statistics are reported alongside the full confusion matrix and class counts, since kappa can be misleading when one category dominates the sample, and raw observed percent agreement is reported alongside kappa for the same reason. Per-class precision, recall, and F1 (treating the adjudicated human label as gold) are reported for the Groq judge to expose category-specific failures that an aggregate kappa can hide. The four-category ("categorical") kappa is reported alongside a binary right/wrong collapse ($\{$correct, partial$\}$ vs. $\{$incorrect, hallucinated$\}$), with a prevalence-and-bias-adjusted kappa (PABAK $= 2p_o - 1$) for that binary collapse, since "was the specific label right" and "was the pass/fail call right" are different claims that a single kappa can conflate. A secondary, quota-limited 10-item subsample (the main-response subset of the 30) additionally includes the Gemini judge and a Groq–Gemini inter-judge kappa; this secondary result is reported separately and never pooled with the 30-item primary result.

V. Results

A. Study 1 Results

On the 18-task set, ROUGE-L F1 has mean 0.086 and median 0.0: only 2 of the 18 responses have any lexical overlap with their reference answer at all (F1 of 0.55 and 1.00), and the remaining 16 score exactly 0. That pattern fits a single-call agent, backed only by DuckDuckGo search snippets, failing GAIA's multi-hop lookups outright, rather than producing near-miss phrasing that ROUGE-L would at least partially reward. In other words, ROUGE-L's well-known weakness of penalizing correct paraphrase is not the dominant failure mode in this sample; the agent itself is.

On the 8-task, $k = 3$ SelfCheckGPT subset, mean consistency is 0.125, with 7 of 8 tasks scoring 0.0 and one task scoring 1.0. This SelfCheckGPT implementation compares samples using token-level (Jaccard) lexical overlap, not the original entailment/NLI-based scorer of [5]; near-identical wrong numeric guesses across resamples therefore still register as "consistent," which is why consistency and correctness diverge below. Per-task Bradley–Terry fitting, after the position-bias control described in Section IV-C, produced at least one usable comparison for 5 of the 8 tasks; the remaining 3 tasks yielded zero position-consistent judge verdicts across all three sample pairs, so no ranking could be fit for them. Comparing the 8 SelfCheckGPT consistency scores against the corresponding Study 2 human hallucination labels (Section V-B) shows no simple correspondence: several samples the judge and humans separately labeled hallucinated still scored 0.0 consistency, since near-identical wrong numeric guesses across resamples are scored as consistent by the lexical overlap check even though they are all incorrect. At $n = 8$ this is reported descriptively as a limitation of the consistency signal in this sample, not as a formal negative correlation result.

Limitations of this result. These numbers come from 18 GAIA tasks run through a single agent configuration (Groq Llama 3.3 70B, one web-search tool); they describe how ROUGE-L, Bradley–Terry, and this lexical SelfCheckGPT proxy behaved on that one setup, not a general claim about these methods on other agents, benchmarks, or task distributions.

B. Study 2 Results

Study 2 has a primary result and a secondary result, reported separately and never pooled. The primary result uses the full 30-item sample (two human annotators plus the Groq judge, all labeling all 30 items). The secondary result uses a 10-item subsample (the main-response items only, a subset of the 30) to additionally include a Gemini judge, since Gemini's free-tier quota made labeling all 30 items impractical.

1) Primary Result: Human–Human and Groq–Human Agreement (n=30): The two human annotators independently labeled all 30 items using the four-category rubric. Human–human Cohen's kappa is $\kappa = 0.7047$ (substantial agreement on the Landis–Koch scale), with observed agreement 0.8333 (25 of 30 items agreed); all 5 disagreements were an incorrect-vs.-hallucinated distinction, adjudicated by discussion to produce the human reference standard. This exceeds the paper's pre-registered 0.6 threshold, supporting the rubric as reliable enough to serve as a reference standard.

The Groq-based LLM judge (Llama 3.3 70B via Groq) labeled the same 30 items independently. Groq–human kappa is $\kappa = 0.1327$ (*slight* agreement), with observed agreement 0.4333 (13 of 30). Table I shows why: Groq never outputs the hallucinated (or partial) label in this sample. It matches humans perfectly on correct (2/2) and incorrect (11/11), but all 17 items the human standard adjudicated as hallucinated were labeled incorrect by Groq. Per-class precision/recall/F1 (Table II, human label as gold) makes the same point in a form that is not sensitive to class prevalence: Groq's hallucinated recall and precision are both 0.000, while its incorrect precision is only 0.393 despite 1.000 recall, because every hallucinated item is counted as a false positive for incorrect.

This category-specific failure is invisible if only a binary right/wrong judgment is checked: collapsing $\{$correct, partial$\}$ vs. $\{$incorrect, hallucinated$\}$ gives binary $\kappa = 1.0$ and PABAK $= 1.0$ (perfect agreement), because Groq's every error keeps the item inside the "wrong" bucket. The categorical kappa (0.1327, slight) and the binary kappa (1.0, perfect) are both computed on the same 30 items and are both correct; they simply answer different questions, which is why we report both rather than a single aggregate number.

TABLE I
GROQ JUDGE VS. HUMAN LABELS, PRIMARY SAMPLE ($n = 30$); ROWS = GROQ JUDGE LABEL, COLUMNS = HUMAN ADJUDICATED LABEL

	correct	hallucinated	incorrect
correct	2	0	0
incorrect	0	17	11

TABLE II
PER-CLASS PRECISION/RECALL/F1, GROQ JUDGE VS. HUMAN GOLD ($n = 30$)

Class	Precision	Recall	F1
correct	1.000	1.000	1.000
incorrect	0.393	1.000	0.564
hallucinated	0.000	0.000	0.000

Limitations of this result. This is one Llama-family judge (Groq Llama 3.3 70B), one rubric, and 30 items adjudicated by the paper's two authors; the perfect binary PABAK is a direct consequence of Groq happening not to confuse correct with wrong on this sample, not evidence that the judge is generally reliable at pass/fail calls.

2) Secondary Comparison: 10-Item Subsample with a Second Judge (Gemini): On the 10-item main-response subsample only, human–human kappa is $\kappa = 0.7561$ (substantial, observed 0.9); this differs from the 30-item primary value (0.7047) simply because it is a smaller, non-random subset of the same 30 items, not a separate replication. The Groq judge on this subsample reaches $\kappa = 0.4444$ (moderate, observed 0.8) against the same human standard – higher than its own 30-item primary result, illustrating how much a judge-agreement estimate can swing with sample choice alone at these sizes.

The Gemini-based LLM judge (Gemini 2.0 Flash) labeled the same 10 items independently. Gemini–human kappa is $\kappa = 0.2$ (fair agreement, observed 0.6). Unlike Groq, Gemini uses the partial label, but incorrectly so for 2 items the humans called incorrect, and it too maps both human-labeled hallucinated items in this subsample to incorrect. Groq–Gemini inter-judge kappa is $\kappa = 0.4444$ (moderate, observed 0.8): the two judges agree on simple cases but diverge on nuanced errors, with Gemini's partial usage adding a further source of disagreement beyond the shared hallucinated-collapse failure.

TABLE III
GEMINI JUDGE VS. HUMAN LABELS, SECONDARY SUBSAMPLE ($n = 10$); ROWS = GEMINI JUDGE LABEL, COLUMNS = HUMAN ADJUDICATED LABEL

	correct	partial	incorrect	hallucinated
correct	1	0	0	0
incorrect	0	0	5	2
partial	0	0	2	0

Limitations of this result. At $n = 10$ this secondary comparison is underpowered on its own; it is reported only as a quota-limited robustness check on the 30-item primary finding, not as an independent confirmation with its own statistical power.

VI. DISCUSSION

A. Judge Reliability and the Hallucinated Category

On the 30-item primary sample, the Groq judge tested here shows a systematic failure to distinguish hallucinated (unsupported, fabricated claims) from incorrect (wrong answers): all 17 human-adjudicated hallucinated items were labeled incorrect, giving it 0.000 precision and recall on that class (Table II) even though its aggregate binary right/wrong call was perfect on this sample. On the smaller 10-item secondary subsample, the Gemini judge shows the same collapse plus additional noise from overusing the partial label. We read this as a property of these two specific judge configurations on this rubric and this task set, not as a general claim about hallucinated-vs-incorrect discrimination in LLM judges as a category; a single Llama-family model and a single Gemini model, on one 18-task GAIA subset, cannot establish that. This collapse is nonetheless consequential for anyone considering the same setup, because distinguishing hallucination is precisely the use case that motivates LLM-as-a-judge over simpler keyword overlap.

Scoped to this study's two judges and this rubric, the results suggest that neither is ready to replace human review specifically for the hallucinated-vs-incorrect distinction, while both do reasonably on the coarser binary correct/incorrect call. This direction is consistent with concurrent, larger-scale benchmarks such as JudgeBench [10], which report that strong LLM judges perform only slightly better than random on challenging preference pairs, and AgentRewardBench [12], which finds no single judge performs well across all settings – but those benchmarks test many judges and many task families where this study tests one judge configuration each on one benchmark, so we present the alignment as suggestive corroboration, not as replication.

B. Inter-Judge Consistency and the Role of Multiple Judges

On the 10-item secondary subsample, Groq–Gemini kappa is $\kappa = 0.4444$ (moderate), despite both being deployed on identical items with the same rubric. The two judges use different label distributions: Groq never outputs partial, while Gemini overuses it. This divergence suggests that combining judges (e.g., majority vote or probabilistic aggregation) might help in higher-stakes evaluation, though that design was not tested here and the observation rests on only 10 items.

Human–human agreement is $\kappa = 0.7047$ on the full 30-item primary sample (substantially above both judges' 30-item and 10-item results), providing a baseline: the two human annotators, after calibration and independent blind annotation, agree with each other well beyond either judge's agreement with them. This suggests rubric clarity is not the barrier; the judges' ability to apply the rubric depends on model-specific factors this study did not vary.

C. Study 1 Implications Reconsidered

Study 1's finding that Bradley–Terry and SelfCheckGPT produced low signal on the 8-task, $k = 3$ subset (only 5 of 8 tasks yielded position-consistent pairwise comparisons,

and SelfCheckGPT consistency was near-zero for 7 of 8 tasks) takes on a different meaning in light of Study 2's judge-reliability results. The low Bradley–Terry yield and SelfCheckGPT consistency are not primarily indicators of computational issues but rather reflect the true difficulty of the task: the agent's answers were often wrong outright, making it impossible for consistency-based or pairwise signals to extract meaningful distinctions. Study 2's finding that human–human agreement, while substantial, is not perfect even on a fixed rubric with calibration ($\kappa = 0.7047$, 30 items) suggests that agent evaluation in this domain is genuinely hard, not an artifact of misconfigured algorithms.

D. Implications for Agent Evaluation Practice

Three practical conclusions follow for anyone building on this pipeline, scoped to the single agent configuration and single benchmark studied here.

1) LLM judges alone are not enough for hallucination detection in this setup. If distinguishing hallucinated claims matters for a given use case, automated evaluation needs to pair LLM judges with human spot-checks or an external fact-checking system. A judge that correctly flags wrong answers but mislabels hallucinations buys little over a simpler semantic-similarity baseline.
2) Judge agreement should be checked per rubric category and as both a categorical and a binary statistic, not with one aggregate kappa. Table II and the binary-vs-categorical comparison in Section V-B show that a single number can hide a category-specific failure entirely: this study's judge looked perfect on binary right/wrong and unreliable ($\kappa = 0.1327$) on the 4-class rubric, on the exact same 30 items.
3) Reference metrics still earn their place despite the obvious limits. Study 1 found ROUGE-L near-zero on 16 of 18 tasks, which could read as a sign the metric is useless here. It isn't: a near-zero score still correctly flags that the agent failed, even though it says nothing about how or why. Pairing ROUGE-L with human spot-checks or consistency-based filtering beats leaning on either signal by itself.

E. Limitations and Scope

The study evaluates one agent configuration (Groq Llama 3.3 70B with DuckDuckGo search) on one benchmark (GAIA text-only validation split, 18 main tasks plus 8 tasks with $k = 3$ resamples), with a Gemini secondary judge check on only 10 of the 30 human-labeled items. These are the main threats to external validity: neither the algorithmic results (Study 1) nor the judge-agreement results (Study 2) should be read as claims about LLM judges, Llama-family models, or agent evaluation in general – they describe what happened on this specific configuration and task set, and generalizing further requires testing additional agent configurations, benchmarks, and judge models. The two human annotators are also the paper's authors; calibration, blind labeling, and adjudication reduce but do not remove the risk of shared bias, and two author-annotators do not constitute an independent, unbiased gold standard in the way that recruiting external annotators would. The adjudication rule used to break disagreements (Section IV-D) is stated in the protocol, but individual adjudicated labels are not accompanied by a logged, per-item written rationale, so the human reference standard itself cannot be independently audited item-by-item from the released materials alone.

The study does not test whether fine-tuning, in-context learning, or chain-of-thought prompts could improve judge reliability, nor does it test whether other LLM backbones (GPT-4o, Claude, Qwen, etc.) show different hallucination-detection capabilities. These are important follow-up questions, but they are outside the scope of the current benchmark design.

F. Data and Code Availability

Item-level agent responses, ROUGE-L and SelfCheckGPT scores, both annotators' raw labels, the adjudicated human labels, the Groq and Gemini judge labels, the confusion matrices, the rubric and judge prompts, and the analysis scripts used to compute every statistic reported in this paper (`kappa.py`, `compare_all_labels.py`, `rouge_l.py`, `bradley_terry.py`, `selfcheckgpt.py`) are retained in the project repository accompanying this submission (https://github.com/prashant7738/Agent_Eval_Pipeline), so that every number above can be independently recomputed from the released data.

VII. Conclusion

This study investigated whether automatic metrics and LLM judges provide reliable measurement instruments for evaluating AI agent outputs, focusing on ROUGE-L, Bradley–Terry/Elo ranking, SelfCheckGPT consistency, and Cohen's kappa-based judge–human agreement validation, on one agent configuration and one benchmark.

The main empirical finding is negative but scoped: the two LLM judges tested, Groq Llama 3.3 70B (primary, $n = 30$) and Gemini 2.0 Flash (secondary, $n = 10$), do not reliably distinguish hallucinated claims from incorrect answers when evaluated against human-adjudicated labels on this benchmark. Groq's categorical agreement with humans is only slight on the full 30-item sample ($\kappa = 0.1327$), driven entirely by its never outputting the hallucinated label (0/17 recall on that class), even though its binary right/wrong call on the same items is perfect ($\kappa = 1.0$, PABAK $= 1.0$). On the smaller 10-item secondary subsample, Gemini performs worse against humans ($\kappa = 0.2$, fair) and introduces additional noise through overuse of the partial category, while Groq–Gemini inter-judge agreement is moderate ($\kappa = 0.4444$).

These results are directionally consistent with concurrent judge benchmarks (JudgeBench, AgentRewardBench) that report LLM judges performing poorly on challenging, nuanced distinctions, though those benchmarks test far more judges and task families than the one judge/one benchmark tested here. The study's main methodological point is that treating an LLM judge's aggregate kappa (or its score) as sufficient

evidence, without also checking per-class metrics and a binary-vs-categorical breakdown, can hide a category-specific failure entirely.

The positive findings are narrower: human–human agreement is substantial on the full 30-item sample ($\kappa = 0.7047$), supporting the four-category rubric as clear enough for two trained annotators (who are also this paper's authors) to apply reliably, though this does not by itself establish an unbiased external gold standard. Groq and Gemini both classify items as correct or incorrect reasonably well when the answer is unambiguously right or wrong, but fail precisely where a judge could add the most value: identifying that an answer is wrong in a specific way (hallucinated vs. incorrect).

The practical implication, scoped to this configuration, is that LLM-as-a-judge evaluation should not replace human review on tasks requiring hallucination detection, but may be usable for binary right/wrong assessment once validated with Cohen's kappa, per-class precision/recall/F1, and a confusion matrix on a representative, ideally externally-annotated sample. Reference-based metrics such as ROUGE-L, though limited, retain value as a cheap first-pass filter. Extending this specific finding to LLM judges in general would require testing additional judge models, agent configurations, and benchmarks – a direction for future work, not a claim made by this study.

ACKNOWLEDGMENT

We thank Prof. Rajad Shakya for his guidance and supervision throughout this research. His feedback on the study design and the interpretation of results shaped several parts of this work for the better.

REFERENCES

[1] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. ACL Workshop on Text Summarization Branches Out*, Barcelona, Spain, 2004, pp. 74–81.

[2] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. The method of paired comparisons," *Biometrika*, vol. 39, nos. 3–4, pp. 324–345, 1952.

[3] A. E. Elo, *The Rating of Chessplayers, Past and Present*. New York, NY, USA: Arco, 1978.

[4] L. Zheng *et al.*, "Judging LLM-as-a-judge with MT-Bench and Chatbot Arena," arXiv:2306.05685, 2023.

[5] P. Manakul, A. Liusie, and M. J. F. Gales, "SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models," in *Proc. EMNLP*, 2023, pp. 9004–9017.

[6] D. Li, B. Jiang, L. Huang, A. Beigi, C. Zhao, Z. Tan, A. Bhattacharjee, Y. Jiang, C. Chen, T. Wu, K. Shu, L. Cheng, and H. Liu, "From generation to judgment: Opportunities and challenges of LLM-as-a-judge," arXiv:2411.16594, 2024.

[7] H. Li *et al.*, "LLMs-as-judges: A comprehensive survey on LLM-based evaluation methods," arXiv:2412.05579, 2024.

[8] J. Gu *et al.*, "A survey on LLM-as-a-judge," arXiv:2411.15594, 2025.

[9] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[10] S. Tan, S. Zhuang, K. Montgomery, W. Y. Tang, A. Cuadron, C. Wang, R. A. Popa, and I. Stoica, "JudgeBench: A benchmark for evaluating LLM-based judges," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2025.

[11] M. Zhuge *et al.*, "Agent-as-a-Judge: Evaluate agents with agents," arXiv:2410.10934, 2024.

[12] X. H. Lu, A. Kazemnejad, N. Meade, A. Patel, D. Shin, A. Zambrano, K. Stańczak, P. Shaw, C. J. Pal, and S. Reddy, "AgentRewardBench: Evaluating automatic evaluations of web agent trajectories," arXiv:2504.08942, 2025.

[13] G. Mialon, C. Fourrier, C. Swift, T. Wolf, Y. LeCun, and T. Scialom, "GAIA: A benchmark for general AI assistants," arXiv:2311.12983, 2023.