# Beyond Confidence: Test-Time Scaling for Multi-Turn Search Agents via Retrieval Grounding

**Anonymous ACL submission**

## Abstract

Confidence-based voting aggregates parallel LLM rollouts by weighting each with internal signals such as token log probabilities, and has been actively studied for single-turn reasoning. However, modern LLMs increasingly act as multi-turn search agents that retrieve and condition on external documents. In this paper, we show that confidence-based voting transfers poorly to this multi-turn setting, and identify the underlying failure reason as copy inflation: when retrieved documents are appended to an agent's context, tokens copied from those documents receive systematically inflated log probabilities. This flattens confidence scores within each question and weakens the resulting weighted vote. To address this issue, we propose Retrieval-Grounded Voting (RGV), which scores each rollout by the lexical overlap between its final answer and the documents it retrieved. By computing the signal outside the contaminated context, RGV sidesteps both token log probabilities and additional LLM calls. Across four search-agent benchmarks and five LLMs, RGV consistently outperforms confidence-based voting, with gains of up to +5.4% accuracy and +35% on minority-correct questions, where the correct answer appears in only 1-2 of 8 rollouts.

## 1 Introduction

Confidence-based voting, weighting parallel rollouts by an aggregate of their token-level logprobs, has been widely adopted for test-time scaling in single-turn LLM reasoning (Fu et al., 2025; Taubenfeld et al., 2025), and can outperform majority voting on non-retrieval interactive agent tasks (Wang et al., 2024). But as LLMs are deployed as multi-turn search agents that condition on retrieved documents (Yao et al., 2023; Nakano et al., 2021; Chen et al., 2025), this logprob-based signal transfers poorly. Why confidence voting fails in this setting, and how to address it at the voting layer, remains largely understudied.

Existing work addresses the problem from three perspectives, but each leaves it unresolved. *Confidence-based voters* (Fu et al., 2025; Taubenfeld et al., 2025; Wang et al., 2024) are designed for single-turn LLM settings and offer no diagnosis for search agents. *Trajectory-aware aggregators* (Lee et al., 2026; Li et al., 2025) sidestep the confidence signal by spending an extra LLM call per question (an aggregator on top of the parallel rollouts), adding inference cost. *RL-based recalibration* (Xuan et al., 2026) adjusts confidence via RL fine-tuning but leaves the source of miscalibration in search agents largely unexplained.

An ideal treatment should first *diagnose* why confidence signals degrade in search agents, then leverage that diagnosis to design an efficient, broadly applicable voting-layer solution.

To address this gap, we make two contributions. *First*, we identify and quantify a mechanism we call *copy-inflation*: once retrieved documents are appended to the agent's context, copied tokens receive inflated logprobs, compressing confidence scores within a question so that weighted majority voting collapses to simple majority (§3). *Second*, the diagnosis points to a solution: read the voting signal from *outside the contaminated context*. We propose **Retrieval-Grounded Voting (RGV)**: weight each rollout by the lexical overlap between its answer prose and the documents it retrieved (§4). Because the retrieval log is environment-derived, the signal is independent of the model's hidden state and requires no logprobs, no fine-tuning, and no extra LLM calls.

Across four search agent benchmarks and five LLMs, RGV beats logprob-based confidence voting (DeepConf; Fu et al., 2025), by up to $+5.4\%$ accuracy and $+35\%$ on minority-correct questions. RGV at four rollouts already matches DeepConf at eight, halving the rollout budget at equal accuracy. The gains are robust to the choice of overlap metric and add negligible CPU cost.

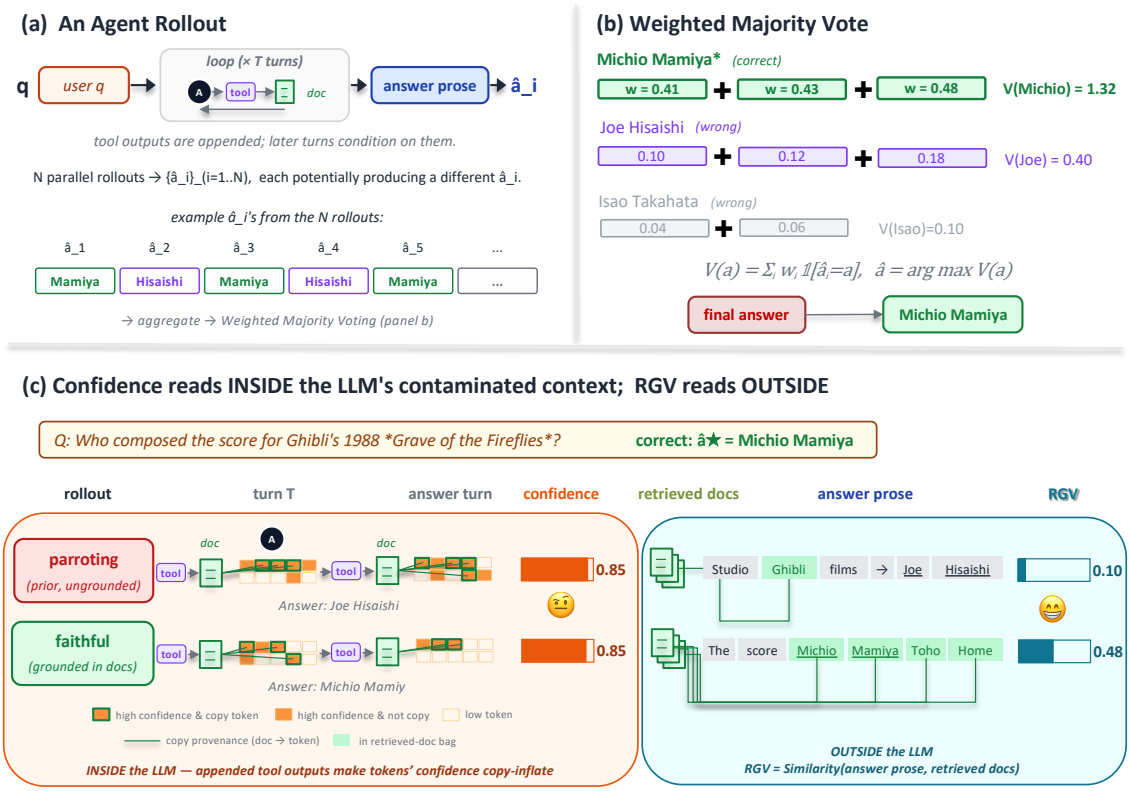



Figure 1: **Overview.** (a) Each rollout is one trajectory of the ReAct-style search agent, looping over tool calls (search, page visit) and emitting a predicted answer; $N$ parallel rollouts may differ. (b) The weighted vote sums per-rollout weights over predicted answers; argmax wins. (c) Confidence-based voting weights each rollout by its token logprobs, inflated when the model parrots context. RGV (ours) weights each rollout by the lexical overlap between its answer prose and the retrieved documents—a signal read outside the contaminated context.

## 2 Background

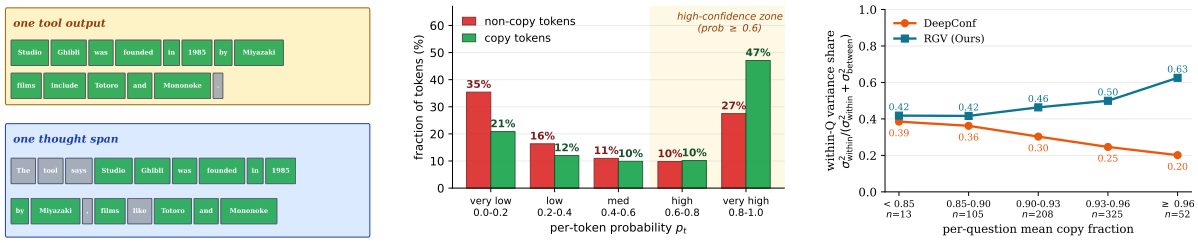
**Multi-turn search agent.** We study ReAct-style agents (Yao et al., 2023; Schick et al., 2023) whose tools retrieve text documents and whose final answer is derived from the retrieved content. We call these *multi-turn search agents*, or *search agents* for short. Such an agent interleaves reasoning and tool calls under a budget of $T$ turns. With context $c_0 = q$, at turn $k$ the policy $\pi_\theta$ samples $(u_k, x_k) \sim \pi_\theta(\cdot \mid c_{k-1})$, where $u_k$ is the model's thought span at turn $k$ and $x_k$ is either a tool invocation (e.g. `search` to query a search engine, or `visit` to fetch a web page) or the final answer. On a tool call the environment returns an observation $o_k$ and the context grows by $c_k = c_{k-1} \| u_k \| x_k \| o_k$. Retrieved snippets are thus appended to the same context the model conditions on at later turns.

**Rollout.** A *rollout* $t_i$ is one trajectory of this loop, terminating on a final answer or budget exhaustion. We sample $N$ rollouts $\{t_1, \ldots, t_N\}$ in parallel from $q$. The rollout's record contains every tool call and its response, every thought span the model emitted, and a final *answer turn* whose output is one last thought span, followed by the predicted answer string. Three names from this record recur below: the predicted answer $\hat{a}_i$, the *answer prose* $P_i$ (the final answer the model produces on the answer turn, excluding its internal chain-of-thought), and the retrieval log $\mathcal{D}_i = \{d_1, \ldots, d_{m_i}\}$ (the documents the agent fetched).

**Weighted majority vote.** The $N$ rollouts may give different answers, so a voting rule returns a single final answer. Each rollout is scored by a non-negative weight $w_i$, and the chosen answer is the one whose rollouts carry the largest total weight:

$$V(a) = \sum_{i=1}^{N} w_i \, \mathbb{I}(\hat{a}_i = a), \qquad \hat{a} = \arg\max_a V(a).$$

Different choices of $w_i$ give different methods. *Simple majority* ($w_i \equiv 1$) just counts rollouts per answer. *DeepConf* (Fu et al., 2025), our confidence baseline, reduces per-token confidences

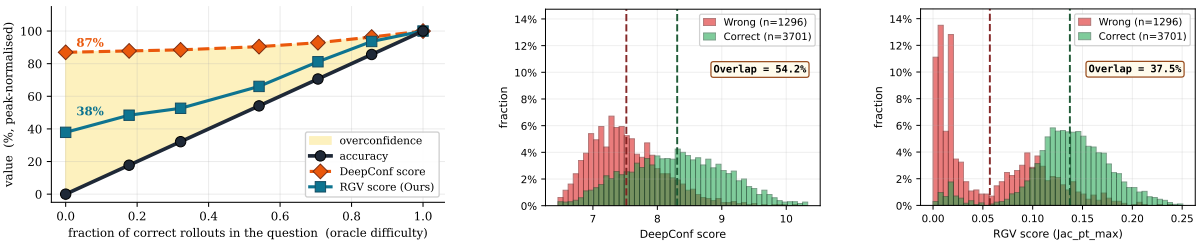



(a) **What is a copy token?** A token whose surface form is a substring of some retrieved doc (green).

(b) **Copy tokens carry inflated confidence.** Per-token probability: copy vs. non-copy.

(c) **Copy-heaviness flattens DeepConf within a question.** Within-$Q$ score-variance share vs. per-$Q$ copy fraction.

Figure 2: **Copy-inflation.** (BrowseComp-Plus, Tongyi-DeepResearch) (a) Defines the copy-token primitive: the doc is already in context, and the model parrots the tool outputs. (b) On generated tokens, copy tokens cluster at prob $\geq$0.6 (57% vs. 37% for non-copy), a $+0.50$ mean logprob gap. (c) The $y$-axis is the within-question share of total DeepConf score variance, i.e. how much of the score's spread comes from differences among rollouts of the *same* question. As per-$Q$ mean copy fraction grows, DeepConf's share drops $0.39 \rightarrow 0.20$ (same-$Q$ rollouts get nearly the same score, so weighted voting cannot separate them); for comparison, RGV (our method, §4) rises $0.42 \rightarrow 0.63$. Median rollout copies 93% of its content tokens; 85% of questions have mean copy fraction $\geq$0.90.




(a) **DeepConf is overconfident.** Mean per-question DeepConf score stays at 87% of peak even on questions where every rollout is wrong.

(b) **DeepConf.** Per-rollout score density split by judge correctness; the two populations overlap heavily.

(c) **RGV (Ours; §4).** Same split as (b), scored by our method (overlap 37.5%).

Figure 3: **Consequence at the rollout level** (BrowseComp-Plus, Tongyi-DeepResearch). (a) Stratifying questions by oracle difficulty (fraction of correct rollouts in the question) reveals DeepConf is nearly flat: it gives impossible questions (87% of peak) almost the same confidence it gives easy ones. (b) DeepConf's per-rollout correct/wrong distributions overlap heavily (54.2%). (c) For comparison: RGV (our method, §4) on the same split.

$C_t = -\frac{1}{k}\sum_{j=1}^{k} \log p_j$ (top-$k$ averaged log-prob, negated; higher means more peaked) over the tokens the model produces during the rollout to a single $w_i$ via a sliding-window aggregation. We use the *Lowest-Group* reduction (sliding-window minimum) at window size $W$=1024 as the headline; the full sweep over reductions and window sizes is in Appendix F. Our method (§4) replaces $w_i$ with a direct grounding check between the answer prose $P_i$ and the retrieved docs $\mathcal{D}_i$.

## 3 Motivation

When the only input is the question, DeepConf (Fu et al., 2025), a logprob-based confidence voter (§2) is well-motivated: peaked next-token posteriors correlate with correctness (Kadavath et al., 2022; Tian et al., 2023). In a search agent, however, that correlation breaks down because the tokens DeepConf reads are systematically inflated by the retrieved docs appended to the context. We document this on BrowseComp-Plus rollouts from Tongyi-DeepResearch in three observations (Fig. 2) and trace the consequence at the rollout level (Fig. 3).

**Tool outputs in context inflate logprobs.** A search agent appends every retrieved snippet and visited page to its context window; when it later writes prose, any token it copies from those appended documents has very high $p(t_k \mid \text{context})$ because the document is right there. The token's logprob says the *copy step* is well-calibrated; it does not say the *copied token* is the right answer. Confidence-based voting cannot tell the difference between a rollout that copied the right entity and one that copied an irrelevant one.

*(i) Copy tokens carry inflated confidence* (Fig. 2b). Recall (§2) that DeepConf reads the tokens the model produces during the rollout. For

each such token we check whether its surface form is a substring of any retrieved doc the agent fetched on this rollout; copy tokens have a mean logprob $+0.50$ nats higher than non-copy tokens (730k tokens from a subset of rollouts where per-token logprobs were re-collected). The signal DeepConf reads is thus contaminated by the appended tool outputs regardless of what those tools returned. The gap is robust to the choice of copy-token definition (Appendix G).

*(ii) At the rollout level, copy-inflation flattens DeepConf scores within a question* (Fig. 2c). We measure DeepConf's *within-question share* of total score variance, i.e. how much of the score's spread comes from differences *among rollouts of the same question*, rather than between questions. Weighted majority voting needs this within-question spread to prefer one rollout over another; if it vanishes, the vote degenerates into plain simple-majority. As copy fraction grows, DeepConf's within-question share shrinks $0.39 \rightarrow 0.20$: same-question rollouts get nearly the same DeepConf score and the weight tiebreaker disappears, precisely on the copy-heavy questions where rollouts disagree most.

*(iii) And this is not an edge case; copy ratios are very high*. Across our rollouts, the median copies $93\%$ of its content tokens from the docs it retrieved, $81\%$ copy at least $90\%$, and $85\%$ of questions have a mean copy fraction $\geq 0.90$ (Fig. 2c, $x$-axis). The two failure modes above therefore apply to nearly the entire benchmark, not a corner case. While copy-inflation can in principle arise whenever tool outputs enter the agent's context; multi-turn search agents are the dominant regime.

**Consequence: the weighted vote no longer tracks correctness.** Figure 3 makes the rollout-level consequence concrete. *(a)* Stratifying questions by oracle difficulty (the fraction of their $N$ rollouts that turn out correct), DeepConf's mean per-question score remains at $87\%$ of its peak even on questions where *every* rollout is wrong: DeepConf cannot tell an impossible question from a solvable one, so the cluster-weighted vote inherits this miscalibration at the voting layer; The pattern replicates across every benchmark and model we test (Appendix D). *(b)* The same pattern holds at the rollout level: DeepConf's per-rollout score distributions for correct vs. wrong rollouts overlap heavily (overlap $=54.2\%$), as copy-inflation lifts wrong rollouts into the same high-confidence zone as correct ones, leaving the confidence weight nothing to discriminate.

**The principle.** The shared cause of failure modes is that the voting signal is computed *inside* the contaminated context, not outside it. Robust voting therefore requires a signal whose measurement target is independent of the model's hidden state; something derived from outside the model. The next section turns this principle into a method.

## 4 From principle to method: Retrieval-Grounded Voting

To address the copy-inflation problem identified in §3, we propose reading the voting signal from *outside the contaminated context*, specifically, from the retrieval log. The key intuition, shared with faithfulness evaluation in summarisation and RAG (Maynez et al., 2020; Min et al., 2023; Es et al., 2024), is that a rollout is more trustworthy when its final answer is lexically anchored in the documents it retrieved.

**The RGV score.** For each rollout $i$ on a given question, we have its answer prose $P_i$ (defined in §2: the answer turn's output text) and the *set* $\mathcal{D}_i = \{d_1, \ldots, d_{m_i}\}$ of documents the agent retrieved during that rollout (one $d_j$ per `search` or `visit` call). We map a text span $X$ to a token set $\mathcal{T}(X)$ using a fixed tokenisation/normalisation rule (Appendix A). The rollout's RGV weight is the maximum *prose-recall*, the fraction of the answer-prose tokens that are anchored in some retrieved document,

$$w_i^{\mathrm{RGV}} = \max_{d \in \mathcal{D}_i} \frac{|\mathcal{T}(P_i) \cap \mathcal{T}(d)|}{|\mathcal{T}(P_i)|}.$$

**Why max-over-docs.** We score against the single best-matching document rather than the union of all retrieved documents to avoid dilution: as the retrieval bag grows, an irrelevant-doc union washes out a strong match to the truly supporting passage. This "take an extreme, not an average" design mirrors local-extreme confidence reductions in DeepConf (Fu et al., 2025), which prefer the most-uncertain window-group over an average across all windows.

**Why the prose-side denominator.** With the numerator fixed at $|\mathcal{T}(P_i) \cap \mathcal{T}(d)|$, the design choice is the denominator. Symmetric Jaccard (Jaccard, 1912) uses $|\mathcal{T}(P_i) \cup \mathcal{T}(d)|$, which over-penalises when the retrieved doc is long, exactly the regime we target. Normalising by $|\mathcal{T}(P_i)|$ keeps the score

length-invariant on the doc side: a well-anchored answer is not punished for retrieving a long supporting passage. Other overlap functions (Lin, 2004; Robertson and Zaragoza, 2009) yield similar headline numbers (Appendix E); the gain comes from the signal source, not the choice of denominator.

**Voting rule.** We plug $w_i = w_i^{\mathrm{RGV}}$ into the weighted majority vote of §2; answer strings are clustered after a light normalisation (lowercase, whitespace collapse, leading-article strip) and the cluster with the largest summed weight wins.

**What is and is not computed inside the model's context.** The token set $\mathcal{T}(P_i)$ is, of course, generated by the model. The token set $\mathcal{T}(d)$ is read directly off the retrieval log: it is what the environment returned, regardless of how the model used it. Even though the docs were appended to the model's context, their token set as a *measurement target* does not depend on the model's hidden state, and a rollout cannot inflate its RGV weight through verbosity or self-assuredness; RGV asks only whether the answer prose is anchored in the documents the rollout actually retrieved.

## 5 Experimental setup

**Benchmarks.** We evaluate on four multi-turn search agent benchmarks where the agent gathers evidence from retrieved documents and derives a grounded answer: BrowseComp-Plus (Chen et al., 2025) (fixed-corpus retrieval), BrowseComp (Wei et al., 2025) (open-web browsing), GAIA (Mialon et al., 2023) (general-assistant tasks dominated by web search), and FRAMES (Krishna et al., 2025) (multi-hop retrieval requiring synthesis across multiple Wikipedia articles). For each benchmark we randomly sample up to 150 questions, held constant across models.

**Models.** We evaluate five LLMs across the four benchmarks above: gpt-oss-120b (OpenAI et al., 2025), MiniMax-M2.7 (MiniMax et al., 2025), GLM-5.1 (GLM-5-Team et al., 2026), Kimi-K2.5 (Team et al., 2026a), and Tongyi-DeepResearch (Team et al., 2026b). The resulting benchmark $\times$ model coverage is reported in Table 1.

**Rollout protocol.** The agent is given two tools per benchmark: `search` (issue a text query, receive ranked snippets) and `get_doc` (fetch a full document by ID) on BrowseComp-Plus (fixed-corpus retrieval), and `search` and `visit` (fetch and render a URL) on the other three benchmarks. We sample $N{=}8$ rollouts per question. Full hyperparameters, tool truncation rules, and the force-final / recovery procedure are in Appendix A.

**Judge.** A single judge is used across all datasets: Qwen3-32B at temperature 0, applied with the official BrowseComp-Plus grading template (Chen et al., 2025). The judge sees only the question, the gold answer, and the rollout's final answer; it does not see voting outcomes or other rollouts.

**Voting protocol.** All voting methods plug a per-rollout weight $w_i$ into the weighted majority vote of §2; answer strings are clustered under the strict normalisation of §4. Methods compared: *Simple-majority* (SM, $w_i \equiv 1$); *DeepConf* (DC; Lowest-Group reduction with window $W{=}1024$, chosen on the full reduction $\times$ window-size grid in Appendix F); *RGV* (§4). Two reference columns in Table 1 are non-voting: *single* = mean per-rollout accuracy, and *oracle* = any-rollout-correct upper bound.

## 6 Results

Table 1 shows that RGV is best in all 20 benchmark$\times$model cells. The pattern is consistent with the copy-inflation story of §3: whenever confidence weights lose within-question spread, grounding is the remaining discriminator.

Gains are largest where voting is hardest: low single-rollout accuracy (e.g. $+5.4\%$ on BrowseComp-Plus Tongyi at $51.6\%$ single) and noisy open-web retrieval (BrowseComp, up to $+6.2\%$). They shrink where majority already saturates (FRAMES; $\geq$ $84\%$ single), but remain nonzero even when DeepConf matches simple majority (e.g. GLM-5.1 on FRAMES: $90.0\%$ vs. $90.7\%$), suggesting that lexical grounding adds information beyond both count (simple-majority votes) and logprob confidence. All margins are stable under within-cell 3-fold splits (Appendix B).

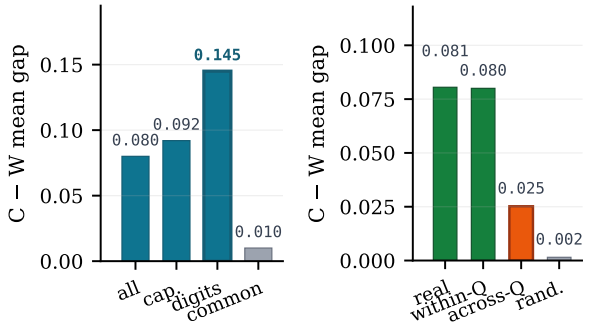
For the mechanism and ablation analyses that follow (§6.1 – §6.3), we anchor on Tongyi-DeepResearch on BrowseComp-Plus ($N{=}8$, 830 questions), where retrieval comes from a fixed corpus (no search-engine noise) and the agent is representative of the class we target.

### 6.1 Is the signal genuine grounding?

§4 interprets overlap as grounding, but a spurious alternative is that it rewards verbosity or common

| Dataset | Model | Single Avg. | Simple Maj. | DeepConf | **RGV (Ours)** | Oracle |
|---|---|---|---|---|---|---|
| BrowseComp-Plus | Tongyi-DeepResearch | 51.6 | 62.0 | 65.7 | **71.1** | 74.2 |
| | OSS-120B | 47.3 | 53.3 | 54.7 | **56.0** | 66.7 |
| | MiniMax-M2.7 | 67.8 | 75.3 | 75.3 | **78.7** | 83.3 |
| | Kimi-K2.5 | 80.0 | 80.7 | 80.7 | **81.3** | 88.7 |
| | GLM-5.1 | 74.9 | 80.7 | 80.7 | **82.0** | 90.0 |
| GAIA | Tongyi-DeepResearch | 70.0 | 79.6 | 79.6 | **80.6** | 92.2 |
| | OSS-120B | 60.4 | 69.9 | 68.9 | **71.8** | 83.5 |
| | MiniMax-M2.7 | 80.3 | 87.4 | 88.3 | **93.2** | 97.1 |
| | Kimi-K2.5 | 73.1 | 78.6 | 79.6 | **81.6** | 92.2 |
| | GLM-5.1 | 83.3 | 87.4 | 87.4 | **89.3** | 92.2 |
| BrowseComp | Tongyi-DeepResearch | 41.2 | 52.0 | 52.7 | **54.0** | 68.7 |
| | OSS-120B | 22.7 | 30.0 | 29.3 | **34.0** | 41.3 |
| | MiniMax-M2.7 | 37.1 | 40.7 | 46.0 | **50.7** | 59.3 |
| | Kimi-K2.5 | 35.0 | 52.8 | 56.6 | **58.5** | 73.6 |
| | GLM-5.1 | 57.8 | 72.6 | 73.7 | **78.9** | 92.6 |
| FRAMES | Tongyi-DeepResearch | 84.0 | 87.3 | 88.0 | **88.0** | 94.7 |
| | OSS-120B | 78.7 | 82.0 | 82.7 | **85.3** | 91.3 |
| | MiniMax-M2.7 | 86.5 | 88.7 | 89.3 | **89.3** | 94.0 |
| | Kimi-K2.5 | 85.7 | 88.0 | 88.7 | **89.3** | 97.3 |
| | GLM-5.1 | 88.3 | 90.0 | 90.0 | **90.7** | 98.0 |

Table 1: **Headline voting accuracies (%)** across 4 multi-turn search agent benchmarks $\times$ 5 LLMs, $N$=8 rollouts per question.




(a) rare-entity matching (b) null controls

Figure 4: **Why RGV works** (BrowseComp-Plus Tongyi-DeepResearch). $y$-axis: C–W mean gap (mean RGV score of correct rollouts minus that of wrong rollouts). (a) Rare-entity tokens carry the signal. (b) The signal survives a within-Q doc shuffle, collapses 3$\times$ across questions, and 53$\times$ under a random-token null.

tokens. If so, RGV would succeed by accident, and overlap could not support the copy-inflation account of §3. We therefore test whether the score reflects genuine retrieval grounding.

At the rollout level, RGV separates judge-correct from judge-wrong trajectories better than DeepConf (ROC AUC (Fawcett, 2006) 0.908 vs. 0.837; +0.071). To explain *why*, we analyse (i) which token classes carry the gap, (ii) controlled shuffles and nulls, and (iii) correlation with an independent retrieval-quality measure.

**Rare-entity matching drives the signal.** Figure 4(a) decomposes the *C–W mean gap*, the difference in mean RGV score between judge-correct (C) and judge-wrong (W) rollouts, by token class. Digit tokens (years, codes, identifiers) give the largest gap (0.145), capitalised entity names follow (0.092), and common content words contribute little (0.010). The pattern is intuitive: common words appear in every rollout's prose regardless of correctness, so they carry no discriminative power; rare entities appear only when the rollout actually retrieved the relevant document. This aligns with the rare-token weighting in classical IR (Robertson and Zaragoza, 2009) and atomic-faithfulness metrics (Min et al., 2023).

**Null controls confirm question-specific grounding.** To distinguish genuine grounding from incidental overlap, we construct three counterfactuals (Figure 4(b)). The C–W mean gap is 0.080. (i) A *within-question* doc shuffle (re-pair each rollout's prose with another rollout's docs from the same question) preserves the gap at 0.080, since same-question rollouts share relevant retrievals. (ii) An *across-question* shuffle collapses the gap to 0.025 (3$\times$ drop). (iii) A length-matched random-token null collapses it to 0.002 (53$\times$). Together, they rule out the two most plausible confounds: (ii) shows the signal is not generic text similarity but depends on the question-document match; (iii) shows it is

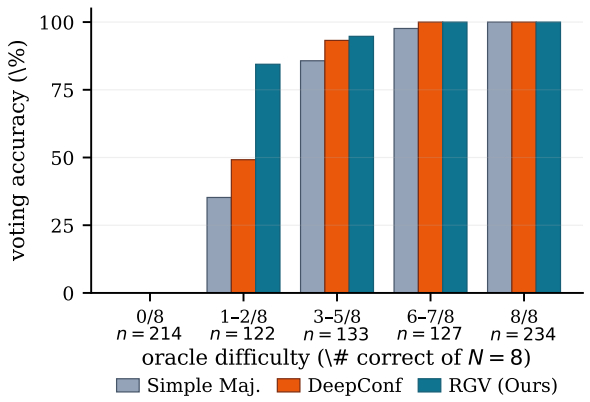



Figure 5: **Voting accuracy by question difficulty** (# correct of $N$=8) on BrowseComp-Plus -DeepResearch. Most of the gap concentrates on the $1-2/8$ bucket.

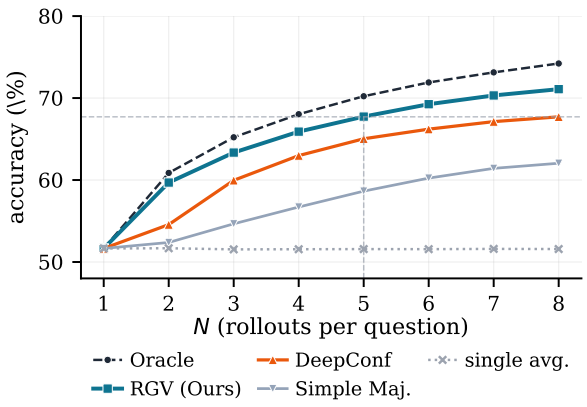



Figure 6: **Accuracy vs. rollout budget** $N$ on BrowseComp-Plus with Tongyi-DeepResearch. RGV is uniformly above DeepConf.

not an artifact of answer length.

**Gold-document validation.** BrowseComp-Plus provides gold supporting documents per question (mean $|\mathcal{G}|$=2.9). For each rollout we compute gold recall, $|\mathcal{D}_i \cap \mathcal{G}|/|\mathcal{G}|$. Judge-correct rollouts have mean gold recall 0.94, vs. 0.31 for judge-wrong (Pearson $r$=0.72). RGV scores correlate with gold recall at $r$=0.57, vs. 0.36 for DeepConf. In other words, a high RGV score is a reliable indicator that the rollout found the right documents, not merely that it produced verbose prose. This closes the validation loop: RGV works because it measures retrieval quality, the very property that copy-inflation prevents DeepConf from reading.

### 6.2 Where the advantage concentrates

The copy-inflation analysis of §3 makes a natural prediction: because copy-inflation compresses DeepConf's within-question spread on exactly the questions where rollouts disagree, RGV's advantage should concentrate on those same hard-to-vote questions. We test this prediction directly.

Figure 5 stratifies the 830 questions by oracle difficulty (the number of the eight rollouts that the judge marks correct). On easy questions ($\geq 3/8$ correct) all methods converge. Most of the accuracy gap between RGV and DeepConf concentrates on the 122 *minority-correct* ($1-2/8$) questions, where one or two of eight rollouts found the right answer but the majority did not: RGV scores 84.4% vs. DeepConf's 49.2% (+35.2%). DeepConf fails here because copy-inflation (§3) gives the wrong majority nearly the same confidence as the correct minority; RGV breaks the tie through grounding. This is also the bucket where the gold-doc recall gap of §6.1 is largest (RGV-argmax 84% vs. DeepConf 71%, +13%), confirming that the voting advantage traces back to retrieval quality. RGV's value is concentrated precisely where it is needed: the ambiguous questions where rollouts disagree and the majority answer may be wrong.

### 6.3 Scaling, cost, and design choices

Two practical questions remain: does RGV's advantage persist under tighter rollout budgets, and is the max-over-docs design of §4 the right choice?

**Scaling with $N$.** Sub-sampling the eight rollouts to $N \in \{1, \ldots, 8\}$ at 60 random partitions per $N$ (Figure 6), RGV at $N$=4 (65.9%) already matches DeepConf at $N$=8 (65.7%): the same accuracy at half the rollout cost. The gap grows monotonically from +3.4% at $N$=2 to +5.4% at $N$=8, meaning RGV benefits more, not less, from additional rollouts. This is the regime where test-time scaling (Snell et al., 2024; Brown et al., 2024) is most relevant, and where a better voting rule yields compounding returns.

**Cost.** RGV uses only the emitted prose and the retrieved snippets already in the trajectory: no extra call, no logprobs, ~0.3 ms per rollout on one CPU thread. By contrast, DeepConf requires per-token logprobs, and aggregator methods (Lee et al., 2026; Chen et al., 2023) add at least one rollout-equivalent of inference. When combined with the iso-accuracy scaling above, RGV offers a favourable cost–accuracy trade-off.

**Max-over-docs ablation.** §4 argues for max over the retrieved-document set to avoid dilution from irrelevant documents. Table 2 validates this: max consistently leads, though all four reduc-

| Reduction | Accuracy (%) |
|---|---|
| Simple Majority (no RGV) | 62.0 |
| $\min_d$ | 69.3 |
| $\text{mean}_d$ | 70.5 |
| $\max_d - \min_d$ | 70.5 |
| $\max_d$ (headline RGV) | **71.1** |

Table 2: **Doc-set reduction ablation** ($N$=8, BrowseComp-Plus, Tongyi-DeepResearch).

tions beat simple majority by $+7$ to $+9\%$. The fact that even $\min$ (the worst-matching document) still yields a $+7.3\%$ gain over simple majority is revealing: the bulk of RGV's advantage comes from the *signal source* itself, i.e. reading prose-vs-retrieval overlap outside the contaminated context. The choice of reduction is secondary; $\max$ adds a further $+0.6$ to $+1.8\%$ by focusing on the single most relevant document and avoiding dilution from irrelevant retrievals.

### 6.4 Error analysis

RGV selects the correct answer on 590 of 616 questions where a correct rollout exists (95.8%). The 26 failures (4.2%) follow a well-characterised pattern: in 24 cases a wrong rollout retrieves documents about the right *topic* but draws the wrong *conclusion*, receiving a higher grounding score because it echoes topically relevant—but factually incorrect—evidence. We call this *well-grounded but wrong*. This residual is small relative to the $+5.4\%$ overall gain; it concentrates on high-diversity questions (mean 4.9 unique clusters vs. 4.0) where many plausible answers compete. Appendix J contrasts three representative successes with three failure cases.

## 7 Related work

**Confidence-based aggregation for single-turn reasoning.** Sample-and-aggregate test-time compute (Wang et al., 2022; Snell et al., 2024; Brown et al., 2024; Wang et al., 2025) typically weights the vote by a confidence proxy: self-reported confidence (Wang et al., 2024; Taubenfeld et al., 2025), sliding-window token logprobs (Fu et al., 2025), an extra LLM judge (Chen et al., 2023; Thirukovalluru et al., 2024), or saturation-based budget (Aggarwal et al., 2023). These assume the question is the only conditioning, with no tool outputs in context.

**The breakdown in multi-turn tool-using agents.** Within the broader class of tool-using agents, we focus on search agents, the subclass where retrieved documents dominate the context. Recent work reports that confidence transfers poorly to this setting, but examines a different signal class. Xuan et al. (2026) document a *confidence dichotomy*—evidence tools induce overconfidence in *verbalised* confidence and recalibrate it via RL fine-tuning; BrowseConf (Ou et al., 2025) uses verbalised confidence as a retry trigger. Both target the verbalised channel rather than the token-logprob signal used by logit-based voters, and neither operates at the voting layer. Wang et al. (2024) soften majority voting with token logprobs on interactive tasks without retrieval. Broader miscalibration accounts (Kadavath et al., 2022; Tian et al., 2023; Xiong et al., 2024; Kuhn et al., 2023) share our framing; our contribution is to address the mechanism at the voting layer without retraining or a second model.

**Trajectory-aware aggregators.** A complementary thread adds an extra model: AggAgent (Lee et al., 2026) aggregates parallel trajectories; ParallelMuse (Li et al., 2025) compresses partial ones for an aggregator; PRM-style verifiers (Cobbe et al., 2021; Lightman et al., 2023) and self-correction agents (Shinn et al., 2023; Zhou et al., 2024) score intermediate steps. These help but add at least one rollout-equivalent of inference; RGV adds none.

**Faithfulness and grounding metrics.** RGV's lexical-overlap signal is a voting-layer use of a familiar intuition from summarisation and RAG evaluation (Maynez et al., 2020; Lin, 2004; Min et al., 2023; Es et al., 2024; Gao et al., 2024). Those metrics are evaluated post-hoc against a single output; we use the same notion of grounding to *choose among* parallel rollouts at inference time.

## 8 Conclusion

We identified copy-inflation, the mechanism by which retrieved documents inflate logprob-based confidence in search agents, and proposed RGV, which reads the voting signal from *outside the contaminated context*. Across four benchmarks and five LLMs, RGV outperforms both simple majority and confidence-based voting while adding minimal cost. Our findings suggest that when retrieved evidence contaminates the model context, aggregation signals drawn from environment-derived records offer a more reliable foundation than internal model signals. We believe this direction may extend beyond voting to other settings where context and evidence are no longer separable.

## Limitations

We focus on multi-turn search agents where retrieved documents are the primary evidence source. Agents that rely on non-retrieval tools (e.g., code interpreters) may generate outputs that are not directly anchored to retrieved passages, and may require different grounding signals. On reasoning-intensive tasks where retrieval mainly supports the answer rather than providing it, RGV might show smaller gains; these settings may need a different design, which we leave to future work. RGV is evaluated only in English, and the lexical signal we use is surface-level; richer grounding checks are a next step. Finally, if the retrieval corpus is adversarially poisoned, RGV is vulnerable, as is any method that trusts the retrieval log.

## References


Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. *Preprint*, arXiv:2305.11860.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *Preprint*, arXiv:2407.21787.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *Preprint*, arXiv:2311.17311.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. 2025. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *Preprint*, arXiv:2508.06600.

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

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Tom Fawcett. 2006. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874. ROC Analysis in Pattern Recognition.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *Preprint*, arXiv:2508.15260.

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

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, and 168 others. 2026. Glm-5: from vibe coding to agentic engineering. *Preprint*, arXiv:2602.15763.

Paul Jaccard. 1912. The distribution of the flora in the alpine zone. *The New Phytologist*, 11(2):37–50.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4745–4759.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Yoonsang Lee, Howard Yen, Xi Ye, and Danqi Chen. 2026. Agentic aggregation for parallel scaling of long-horizon agentic tasks. *Preprint*, arXiv:2604.11753.

Baixuan Li, Dingchu Zhang, Jialong Wu, Wenbiao Yin, Zhengwei Tao, Yida Zhao, Liwen Zhang, Haiyang Shen, Runnan Fang, Pengjun Xie, Jingren Zhou, and Yong Jiang. 2025. Parallelmuse: Agentic parallel thinking for deep information seeking. *Preprint*, arXiv:2510.24698.

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

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1906–1919.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. *Preprint*, arXiv:2311.12983.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *Preprint*, arXiv:2305.14251.

MiniMax, :, Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, Chengjun Xiao, Chengyu Du, Chi Zhang, Chu Qiao, Chunhao Zhang, Chunhui Du, Congchao Guo, and 109 others. 2025. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *Preprint*, arXiv:2506.13585.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, and 1 others. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Litu Ou, Kuan Li, Huifeng Yin, Liwen Zhang, Zhongwang Zhang, Xixi Wu, Rui Ye, Zile Qiao, Pengjun Xie, Jingren Zhou, and Yong Jiang. 2025. Browseconf: Confidence-guided test-time scaling for web agents. *Preprint*, arXiv:2510.23458.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessí, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: language models can teach themselves to use tools. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.

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

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence improves self-consistency in llms. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20090–20111.

Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, and 181 others. 2026a. Kimi k2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, Kuan Li, Liangcai Su, Litu Ou, Liwen Zhang, Pengjun Xie, Rui Ye, Wenbiao Yin, Xinmiao Yu, Xinyu Wang, and 38 others. 2026b. Tongyi deepresearch technical report. *Preprint*, arXiv:2510.24701.

Raghuveer Thirukovalluru, Yukun Huang, and Bhuwan Dhingra. 2024. Atomic self-consistency for better long form generations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12681–12694.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024. Soft self-consistency improves language models agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 287–301.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. 2025. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, volume 2025, pages 33944–33963.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations*, volume 2024, pages 23650–23678.

Weihao Xuan, Qingcheng Zeng, Heli Qi, Yunze Xiao, Junjue Wang, and Naoto Yokoya. 2026. The confidence dichotomy: Analyzing and mitigating miscalibration in tool-use agents. *Preprint*, arXiv:2601.07264.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Denny Zhou, Swaroop Mishra, Huaixiu S Zheng, and 1 others. 2024. Self-discover: Large language models self-compose reasoning structures. *Advances in Neural Information Processing Systems*, 37:126032–126058.

## A Implementation details

**Infrastructure.** Tongyi-DeepResearch rollouts are generated locally on 8× NVIDIA RTX PRO 6000 GPUs; all other models are served via Fireworks AI.[1] Web search is powered by the Serper API[2] and page visits use Crawl4AI.[3]

**Token sets for RGV.** Given a text span $X$, we form the token set $\mathcal{T}(X)$ by (i) lowercasing and Unicode-NFKC-normalising the string; (ii) stripping markdown markup, URLs, code fences, and JSON delimiters; (iii) whitespace-splitting; (iv) removing a small English stopword list of $\sim 130$ function words; and (v) dropping any token whose normalised length is below two characters or whose normalised form is purely punctuation. No lemmatisation or stemming is used. Each document $d_j$ is taken to be the raw text returned by the tool, *not* the model's quoted snippets, so $\mathcal{T}(d)$ reflects what the corpus served and is independent of how the model rendered it.

**Computing the max-over-docs.** For each rollout we cache $\mathcal{T}(P_i)$ once and iterate over $\{d_1, \ldots, d_{m_i}\}$, computing $|\mathcal{T}(P_i) \cap \mathcal{T}(d)|/|\mathcal{T}(P_i)|$ via a single set intersection per document. End-to-end runtime is $O(|P_i| + \sum_j |d_j|)$ and finishes in $\sim 0.3$ ms per rollout on one CPU thread for a typical BrowseComp-Plus rollout.

**Strict answer clustering.** Predicted answer strings are normalised by lowercasing, collapsing whitespace, stripping a leading article (*the/a/an*) and surrounding punctuation, and removing matching outer quotes. Two rollouts join the same cluster iff their normalised strings are byte-equal.

**DeepConf computation.** Per-token confidence is $C_t = -\frac{1}{k}\sum_{j=1}^{k} \log p_j$, the negated mean logprob over the top-$k$ candidates at step $t$. We compute $C_t$ over all of the rollout's generated tokens (reasoning and answer). The headline DeepConf score is the *Lowest-Group* reduction with window size $W{=}1024$: slide a window of 1024 tokens with stride 1 across the generated sequence, compute the mean $C_t$ inside each window, and take the minimum. Other reductions (Bottom-10%, Tail) and window sizes $W \in \{1024, 2048, 4096\}$ are reported in Appendix F.

[1] https://fireworks.ai
[2] https://serper.dev
[3] https://github.com/unclecode/crawl4ai

**Force-final and recovery.** Inside the rollout loop, before each LLM call, we check whether the prompt token count exceeds 85% of the model's context window. If so, the last assistant content is replaced with a truncation marker and a single follow-up call is issued with tools disabled and a markdown final-answer instruction (verbatim in Appendix I). The same force-final is triggered when the agent exhausts `max_iterations` (200), the cumulative output budget (`max_total_output_tokens`, 30K). No post-hoc compressed recovery is performed: if the single force-final call itself fails, the rollout is recorded with status `api_error` and an empty `final_text`, and is treated as `parse_error` by the judge.

**Per-model sampling parameters.** The same protocol is used across all models. Parameters not listed Table 3 take standard defaults. The judge is Qwen3-32B at temperature 0.0.

## B Variance estimation via 3-fold split

The headline Table 1 reports a single mean accuracy per cell over the 150-question sample. To attach uncertainty without changing the protocol, we split each dataset's 150 questions into three disjoint folds of 50 (strata by the random seed already used for sampling: folds use the first 50, next 50, and last 50 qids in their sampling order), recompute each voting method's accuracy on each fold, and report the fold-mean ± fold-standard-deviation in Table 4 below. Folds are fixed across methods so that paired-fold differences are well-defined.
This is a coarse estimator: with three folds the standard deviation is itself noisy. We use it as a stability indicator (does the margin survive a different 50-question slice?) rather than for formal hypothesis tests. The point estimates in Table 1 are computed on all 150 questions.

## C Verbalised confidence as a voting weight

The main text compares RGV against logprob-based confidence (DeepConf), which is the dominant paradigm in the voting literature (Fu et al., 2025; Wang et al., 2022; Snell et al., 2024). A natural question is how *verbalised* confidence—asking the model to self-report a numerical confidence score—performs as an alternative voting weight (Taubenfeld et al., 2025; Xuan et al., 2026; Ou et al., 2025). Verbalised confidence is arguably closer to an "outside-context" signal than raw logprobs,

| Parameter | gpt-oss-120b | MiniMax-M2.7 | GLM-5.1 | Kimi-K2.5 | Tongyi-DeepResearch |
|---|---|---|---|---|---|
| rollout temperature | 0.85 | 0.85 | 0.85 | 0.85 | 0.85 |
| top_p | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| presence penalty | 0.0 | 0.0 | 0.0 | 0.0 | 1.1 |
| max output tokens / turn | 8,192 | 8,192 | 8,192 | 8,192 | 8,192 |
| max total output budget | 30,000 | 30,000 | 30,000 | 30,000 | 30,000 |
| max tool calls | 200 | 200 | 200 | 200 | 200 |
| context window | 131,072 | 196,607 | 202,752 | 256,000 | 131,072 |

Table 3: Per-model sampling parameters. All models use the same system prompt (Appendix I), confidence solicitation protocol, and Qwen3-32B judge at temperature 0.0.

| **Dataset** | **Model** | Simple Maj. | DeepConf | RGV | Oracle |
|---|---|---|---|---|---|
| BrowseComp-Plus | Tongyi-DeepResearch | 62.0±1.1 | 65.7±1.0 | **71.1**±1.9 | 74.2±0.5 |
| | OSS-120B | 53.3±4.7 | 54.7±4.1 | **56.0**±2.8 | 66.7±4.1 |
| | MiniMax-M2.7 | 75.3±6.8 | 75.3±7.4 | **78.7**±8.2 | 83.3±1.9 |
| | Kimi-K2.5 | 80.7±5.0 | 80.7±4.1 | **81.3**±5.7 | 88.7±0.9 |
| | GLM-5.1 | 80.7±4.5 | 80.7±5.2 | **82.0**±4.8 | 90.0±1.6 |
| GAIA | Tongyi-DeepResearch | 79.6±5.8 | 79.6±5.8 | **80.6**±3.9 | 92.2±4.2 |
| | OSS-120B | 69.9±3.1 | 68.9±4.5 | **71.8**±3.8 | 83.5±2.9 |
| | MiniMax-M2.7 | 87.4±3.6 | 88.3±5.0 | **93.2**±2.7 | 97.1±2.4 |
| | Kimi-K2.5 | 78.6±3.7 | 79.6±4.8 | **81.6**±2.1 | 92.2±2.8 |
| | GLM-5.1 | 87.4±0.2 | 87.4±2.4 | **89.3**±2.2 | 92.2±0.0 |
| BrowseComp | Tongyi-DeepResearch | 52.0±2.8 | 52.7±3.8 | **54.0**±2.5 | 68.7±3.4 |
| | OSS-120B | 30.0±0.9 | 29.3±0.9 | **34.0**±1.6 | 41.3±3.4 |
| | MiniMax-M2.7 | 40.7±5.2 | 46.0±4.3 | **50.7**±6.2 | 59.3±3.8 |
| | Kimi-K2.5 | 52.8±10.4 | 56.6±8.1 | **58.5**±7.8 | 73.6±13.3 |
| | GLM-5.1 | 72.6±6.0 | 73.7±5.3 | **78.9**±5.3 | 92.6±0.9 |
| FRAMES | Tongyi-DeepResearch | 87.3±3.1 | 88.0±2.8 | **88.0**±3.5 | 94.7±2.1 |
| | OSS-120B | 82.0±4.7 | 82.7±5.0 | **85.3**±5.7 | 91.3±3.4 |
| | MiniMax-M2.7 | 88.7±1.9 | 89.3±2.8 | **89.3**±2.5 | 94.0±1.6 |
| | Kimi-K2.5 | 88.0±2.8 | 88.7±3.4 | **89.3**±4.7 | 97.3±1.3 |
| | GLM-5.1 | 90.0±2.8 | 90.0±2.8 | **90.7**±1.9 | 98.0±0.9 |

Table 4: Headline accuracies (%) with fold-mean±std under a 3-fold split of each benchmark's question pool.

since it does not directly read per-token probabilities from the contaminated generation; however, the self-assessment is still produced by the same model reasoning under the same context, so it remains susceptible to the overconfidence patterns documented by Xuan et al. (2026). We include it here for completeness as a third signal class, distinct from both the logit-based (DeepConf) and the retrieval-based (RGV) families.

**Verbalised-confidence elicitation.** After each rollout's final answer is produced, the agent is issued one deterministic follow-up call at temperature 0, with tools disabled (`tool_choice="none"`) and `max_tokens=1024`. The prompt asks for a single integer between 0 and 100 representing the verbalised confidence that the just-given answer is correct (verbatim text in Appendix I). The reply is parsed for the last labelled or bare integer in $[0, 100]$ and stored as a per-rollout score; this score is used only in Appendix C.

The protocol elicits a verbalised confidence $v_i \in [0, 100]$ on every rollout (§5, Appendix I). We report it separately from the main-text comparison to keep the signal-class contrast clean (logit-based vs. retrieval-based); this section evaluates it as a voting weight under the same weighted majority rule as the other methods.

Verbalised confidence occupies an intermediate position: it does not directly read per-token logprobs, but the self-assessment is still produced under the same context that contains the retrieved documents. As Xuan et al. (2026) and Ou et al. (2025) also observe, this makes it partially susceptible to the overconfidence patterns of §3, though to a lesser degree than logprob-based signals.

**Combining signals.** The DC+RGV and Verb+RGV columns in Table 5 report a simple combination: within each question, we $z$-normalise each signal (zero mean, unit variance) and form the voting weight

| Dataset | Model | SM | DeepConf | VERB | RGV | DeepConf+RGV | Verb+RGV |
|---|---|---|---|---|---|---|---|
| BrowseComp-Plus | Tongyi-DeepResearch | 62.0 | 65.7 | 68.0 | 71.1 | 70.7 | **71.3** |
| | OSS-120B | 53.3 | 54.7 | 56.7 | 56.0 | 56.0 | **58.0** |
| | MiniMax-M2.7 | 75.3 | 75.3 | 74.0 | **78.7** | 78.0 | 78.0 |
| | Kimi-K2.5 | 80.7 | 80.7 | 80.0 | 81.3 | 81.3 | **82.0** |
| | GLM-5.1 | 80.7 | 80.7 | 82.0 | 82.0 | 82.0 | **82.7** |
| GAIA | Tongyi-DeepResearch | 79.6 | 79.6 | 73.8 | **80.6** | **80.6** | 79.6 |
| | OSS-120B | 69.9 | 68.9 | 71.8 | 71.8 | 71.8 | **72.8** |
| | MiniMax-M2.7 | 87.4 | 88.3 | 91.3 | **93.2** | 92.2 | **93.2** |
| | Kimi-K2.5 | 78.6 | 79.6 | 82.5 | 81.6 | 82.5 | **85.4** |
| | GLM-5.1 | 87.4 | 87.4 | 86.4 | **89.3** | **89.3** | 88.3 |
| BrowseComp | Tongyi-DeepResearch | 52.0 | 52.7 | 53.3 | **54.0** | 53.3 | **54.0** |
| | OSS-120B | 30.0 | 29.3 | **35.3** | 34.0 | 34.0 | 34.7 |
| | MiniMax-M2.7 | 40.7 | 46.0 | 48.0 | 50.7 | **52.7** | 52.0 |
| | Kimi-K2.5 | 52.8 | 56.6 | 60.4 | 58.5 | 60.4 | **62.3** |
| | GLM-5.1 | 72.6 | 73.7 | **78.9** | **78.9** | 74.7 | **78.9** |
| FRAMES | Tongyi-DeepResearch | 87.3 | 88.0 | 87.3 | **88.0** | **88.0** | 87.3 |
| | OSS-120B | 82.0 | 82.7 | 82.7 | **85.3** | 84.7 | 84.0 |
| | MiniMax-M2.7 | 88.7 | 89.3 | 88.0 | **89.3** | **89.3** | 89.3 |
| | Kimi-K2.5 | 88.0 | 88.7 | 84.0 | **89.3** | 88.7 | 88.0 |
| | GLM-5.1 | 90.0 | 90.0 | 90.0 | **90.7** | 90.0 | **90.7** |

Table 5: Voting accuracies (%) including the verbalised-confidence weight (VERB).

$w_i = z(\text{RGV}_i) + \alpha\, z(\text{other}_i)$, where $\alpha$ is selected from $\{0.05, 0.1, 0.15, 0.2, 0.3, 0.5, 1.0\}$ per cell to maximise voting accuracy. Verb+RGV improves over RGV alone in 9 of 20 cells (up to $+3.8\%$ on GAIA Kimi-K2.5 and BrowseComp Kimi-K2.5), suggesting that verbalised confidence captures signal orthogonal to lexical grounding. DC+RGV improves in 3 of 20 cells, indicating that logprob-based confidence is largely subsumed once grounding is accounted for. The Verb+RGV combination is a promising direction for settings where a follow-up confidence call is affordable.

## D Cross-dataset evidence for copy-inflation

The mechanism analysis in §3 uses BrowseComp-Plus with Tongyi-DeepResearch as the deep-dive configuration. To verify that the copy-inflation pattern is not specific to that setting, we replicate the key diagnostic figures across all twenty benchmark×model cells.

Figure 3(a) showed that DeepConf gives impossible questions 87% of its peak score on BrowseComp-Plus with Tongyi. Figure 7 extends this across all twenty configurations. On every combination, impossible questions receive 82–97% of the peak, confirming that copy-inflation is a general property of logprob-based confidence in multi-turn search agents, not an artefact of one benchmark or model.

## E Robustness to alternative lexical-overlap variants

The headline method (§4) uses prose-recall under the strict clustering rule. To check that the gain comes from the signal source rather than the specific overlap function, we swap the per-rollout score $w_i^{\text{RGV}}$ for other members of the set-overlap family (and a few neighbours), keeping everything else fixed (Table 6).

*Definitions.* Jaccard: $|\mathcal{T}(P) \cap \mathcal{T}(d)| / |\mathcal{T}(P) \cup \mathcal{T}(d)|$ (symmetric variant of prose-recall). ROUGE-2: bigram $F_1$ between prose and doc. ROUGE-L: unigram $F_1$ (approximating LCS-based ROUGE-L). BM25: scored as the maximum BM25 score of $P$ against each $d$ under the per-rollout retrieval set as the corpus. TF-IDF: cosine of TF-IDF vectors (same corpus).

## F DeepConf variant grid

Fu et al. (2025) propose three reductions of sliding-window group confidences: *Lowest-Group* (sliding-window minimum, the extreme case of Bottom-$q\%$ as $q \to 0$), *Bottom*-10%, and *Tail*. We compute the per-token confidence as they do, $C_t = -\frac{1}{k}\sum_{j=1}^{k} \log p_j$, and sweep all three reductions across three window sizes $W \in \{1024, 2048, 4096\}$ on the rollout's generated tokens; per-rollout weights are then plugged into the weighted majority vote of §2 (Table 7).

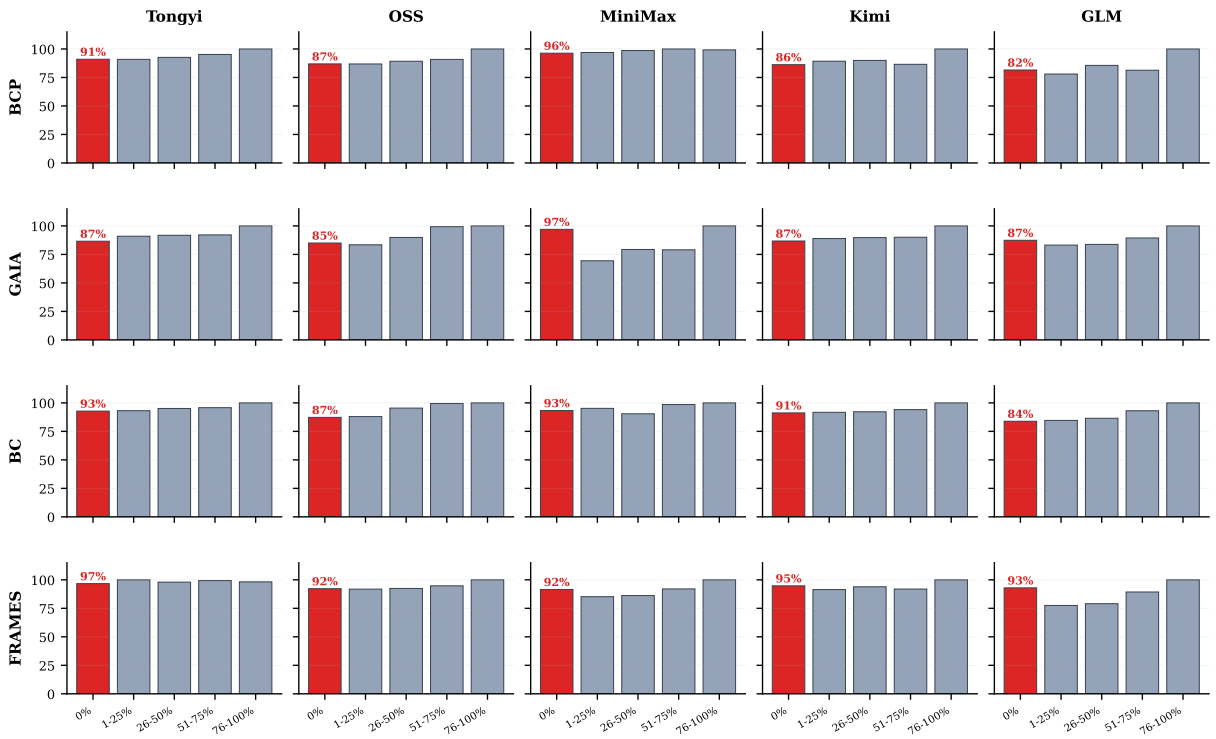



Figure 7: **DeepConf overconfidence across all 20 benchmark×model cells** (cf. Figure 3a). Each panel stratifies questions by oracle difficulty and reports the mean DeepConf score as a percentage of its peak. On every configuration, impossible questions (0% correct, red) receive 82–97% of the peak score. Rows: datasets; columns: models.

## G Robustness of the copy-inflation gap

The headline number in §3 (Fig. 2) is the per-token mean logprob gap between copy and non-copy content tokens: $+0.50$ nats overall, $+0.61$ on judge-correct rollouts, $+0.38$ on judge-wrong ones. To check the gap is not an artefact of any single token-subset choice we re-measure under four alternative rules. All measurements use the same pool of $n$=307 rollouts drawn from the BrowseComp-Plus $\times$ Tongyi-DeepResearch runs (830 questions, 8 rollouts each). The pool is constructed by uniformly sampling 50 questions per rollout seed and keeping those whose token-level statistics and retrieved-doc texts are both available (307/400); the random seed is fixed for reproducibility.

**Common procedure.** For each rollout we form two artefacts. (i) The *retrieved-doc text* $D$: a lowercased, whitespace-collapsed concatenation of everything the search and visit tools returned to the agent. (ii) The *generated-token stream*: every per-token logprob the agent emitted inside its chain-of-thought reasoning span. Each generated token is normalised by stripping subword-piece prefixes, lower-casing, and keeping only alphanumeric characters; tokens that normalise to fewer than two characters are discarded. A token is a *copy token* if its normalised surface is a substring of $D$, else *non-copy*. Substring matching makes the test robust to BPE splits of rare entities.

For a token-subset rule with predicate $\phi(\cdot)$ and weight $\omega(\cdot)$ we pool every kept token across rollouts and report

$$\Delta = \frac{\sum_{t \in \text{copy},\, \phi(t)} \omega(t) \log p(t)}{\sum_{t \in \text{copy},\, \phi(t)} \omega(t)} - \frac{\sum_{t \in \overline{\text{copy}},\, \phi(t)} \omega(t) \log p(t)}{\sum_{t \in \overline{\text{copy}},\, \phi(t)} \omega(t)},$$

the same token-level aggregation as the main-text headline of $+0.50$ nats.

**Variants.**

- **All alnum tokens**: $\phi \equiv \text{True}$, $\omega \equiv 1$. Baseline; reproduces the main-text headline.
- **Stopword-removed**: $\phi(t) = [w_t \notin \mathcal{S}]$ with $\mathcal{S}$ a $\sim$ 120-word English stopword list.
- **IDF-weighted**: $\phi$ as in stopword-removed, $\omega(t) = \log(N/\text{df}(w_t))$ with sample IDF over $N$=307 documents-as-rollouts.

| Dataset | Model | prose-recall | Jaccard | ROUGE-2 | ROUGE-L | BM25 | TF-IDF |
|---|---|---|---|---|---|---|---|
| BrowseComp-Plus | Tongyi-DeepResearch | 71.1 | 71.1 | 70.2 | **71.3** | **71.7** | 69.0 |
| | OSS-120B | 56.0 | 56.0 | **56.7** | 55.3 | 55.3 | 52.7 |
| | MiniMax-M2.7 | **78.7** | 77.3 | 74.0 | 76.0 | 75.3 | 75.3 |
| | Kimi-K2.5 | **81.3** | 80.0 | 79.3 | 79.3 | 77.3 | 78.7 |
| | GLM-5.1 | **82.0** | 79.3 | 80.0 | 80.7 | 79.3 | 78.7 |
| GAIA | Tongyi-DeepResearch | **80.6** | 70.9 | 72.8 | 71.8 | 71.8 | 73.8 |
| | OSS-120B | **71.8** | 69.9 | 68.9 | 68.0 | 68.0 | 67.0 |
| | MiniMax-M2.7 | **93.2** | 91.3 | 91.3 | 91.3 | 91.3 | 89.3 |
| | Kimi-K2.5 | **81.6** | 78.6 | 77.7 | 76.7 | 79.6 | 78.6 |
| | GLM-5.1 | **89.3** | 88.3 | 88.3 | 87.4 | 88.3 | 87.4 |
| BrowseComp | Tongyi-DeepResearch | **54.0** | 53.3 | **54.0** | 52.7 | 50.7 | 52.0 |
| | OSS-120B | **34.0** | 33.3 | 32.7 | 32.7 | 31.3 | 30.7 |
| | MiniMax-M2.7 | **50.7** | 48.0 | 45.3 | 45.3 | 46.0 | 46.0 |
| | Kimi-K2.5 | **58.5** | 52.8 | **58.5** | 52.8 | 50.9 | 52.8 |
| | GLM-5.1 | 78.9 | **82.1** | 77.9 | 76.8 | 69.5 | 81.1 |
| FRAMES | Tongyi-DeepResearch | **88.0** | **88.0** | 86.7 | 87.3 | 86.0 | 86.7 |
| | OSS-120B | **85.3** | 79.3 | 82.0 | 82.0 | 80.7 | 80.0 |
| | MiniMax-M2.7 | **89.3** | **89.3** | 88.0 | 88.7 | 88.0 | 87.3 |
| | Kimi-K2.5 | 88.7 | **89.3** | 87.3 | 88.0 | 86.7 | 87.3 |
| | GLM-5.1 | **90.7** | 90.0 | 89.3 | 90.0 | 89.3 | 88.7 |

Table 6: Voting accuracy (%) across lexical-overlap variants on the main cells. All variants use the same max-over-docs reduction; only $w_i^{\mathrm{RGV}}$ changes. The highest value in each row is bolded.

- **Digits or capitalised-leading**: digit tokens of length $\geq 2$ or BPE-stripped tokens starting uppercase, $\omega \equiv 1$.
- **Length** $\geq 8$: $\phi(t) = [|w_t| \geq 8]$, $\omega \equiv 1$.

All five rules produce a significantly positive gap, confirming that the copy-inflation gap is not a side-effect of any particular token-subset choice. The gap is largest under IDF-weighting, consistent with the visual story that rare, entity-like tokens (the kind retrieval brings into context) are exactly the ones the agent copies.

# H Preliminary results on HLE

Our main evaluation focuses on search agent benchmarks where retrieved documents are the primary evidence source (§5). Here, we report one exploratory cell on Humanity's Last Exam (Phan et al., 2025), an expert-reasoning benchmark where retrieval plays a minor role:

| | Single | SM | DC | **RGV** | Oracle |
|---|---|---|---|---|---|
| OSS-120B | 25.6 | 31.2 | 31.8 | **32.5** | 49.0 |

RGV still outperforms DeepConf (+0.7%), though the margin is smaller than on search-centric benchmarks. This is expected: HLE questions often require domain-expert reasoning beyond what retrieval supplies, so the grounding signal captures only part of the answer quality.

# I Worked examples: prompts, thought spans, and clustering

This appendix shows the exact text the agent and the judge see, plus illustrative excerpts of what a rollout produces. The examples are lightly redacted (line breaks, bracketed abbreviations) for readability; nothing functional is changed.

## I.1 Agent system prompt

Below is the system prompt for BrowseComp-Plus (fixed-corpus retrieval with `search` and `get_doc`). For the open-web benchmarks (BrowseComp, GAIA, FRAMES) the tool names are `search` and `visit` and references to `[docid]` become `[url]`; the rest of the prompt is identical.

```
You are a deep research assistant. Your
core function is to conduct thorough,
multi-source investigations into any
topic. You must handle both broad,
open-domain inquiries and queries
within specialized academic fields. For
every request, synthesize information
from credible, diverse sources to
deliver a comprehensive, accurate, and
objective response.
# Tools
You have access to search and get_doc
tools. Use search to look up snippets;
use get_doc(docid) to read full
documents you find via search. You may
call tools many times until you have
gathered sufficient evidence.
# Final answer format
```

| Dataset | Model | Lowest-Group (min) | | | Bottom-10% | | | Tail | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1024 | 2048 | 4096 | 1024 | 2048 | 4096 | 1024 | 2048 | 4096 |
| BrowseComp-Plus | Tongyi-DeepResearch | **65.7** | 65.4 | 64.8 | **65.7** | 65.5 | 64.9 | 61.8 | 61.9 | 61.4 |
| | OSS-120B | **54.7** | 54.0 | 53.3 | **54.7** | 54.0 | 53.3 | 53.3 | 52.7 | 52.0 |
| | MiniMax-M2.7 | **75.3** | 74.7 | 74.0 | **75.3** | 74.7 | 73.3 | 73.3 | 72.7 | 72.0 |
| | Kimi-K2.5 | **80.7** | 80.0 | 79.3 | **80.7** | 80.0 | 79.3 | 78.0 | 78.0 | 77.3 |
| | GLM-5.1 | **80.7** | 80.0 | 79.3 | **80.7** | 80.0 | 79.3 | 78.7 | 78.0 | 77.3 |
| GAIA | Tongyi-DeepResearch | **79.6** | 78.6 | 77.7 | **79.6** | 78.6 | 77.7 | 76.7 | 76.7 | 75.7 |
| | OSS-120B | **68.9** | 67.0 | 65.0 | **68.9** | 67.0 | 65.0 | 64.1 | 64.1 | 63.1 |
| | MiniMax-M2.7 | **88.3** | 87.4 | 86.4 | **88.3** | 87.4 | 86.4 | 85.4 | 84.5 | 83.5 |
| | Kimi-K2.5 | **79.6** | 78.6 | 77.7 | **79.6** | 78.6 | 77.7 | 76.7 | 75.7 | 74.8 |
| | GLM-5.1 | **87.4** | 86.4 | 85.4 | **87.4** | 86.4 | 85.4 | 85.4 | 84.5 | 83.5 |
| BrowseComp | Tongyi-DeepResearch | **52.7** | 51.3 | 50.0 | **52.7** | 51.3 | 50.0 | 49.3 | 48.7 | 47.3 |
| | OSS-120B | **29.3** | 28.7 | 28.0 | **29.3** | 28.7 | 28.0 | 27.3 | 27.3 | 26.7 |
| | MiniMax-M2.7 | **46.0** | 44.7 | 43.3 | **46.0** | 44.7 | 43.3 | 42.0 | 41.3 | 40.7 |
| | Kimi-K2.5 | **56.6** | 54.7 | 52.8 | **56.6** | 54.7 | 52.8 | 52.8 | 50.9 | 49.1 |
| | GLM-5.1 | **73.7** | 72.6 | 71.6 | **73.7** | 72.6 | 71.6 | 69.5 | 68.4 | 67.4 |
| FRAMES | Tongyi-DeepResearch | **88.0** | 87.3 | 86.7 | **88.0** | 87.3 | 86.7 | 86.0 | 85.3 | 84.7 |
| | OSS-120B | **82.7** | 82.0 | 81.3 | **82.7** | 82.0 | 81.3 | 80.0 | 80.0 | 79.3 |
| | MiniMax-M2.7 | **89.3** | 88.7 | 88.0 | **89.3** | 88.7 | 88.0 | 86.7 | 86.0 | 85.3 |
| | Kimi-K2.5 | **88.7** | 88.0 | 87.3 | **88.7** | 88.0 | 87.3 | 86.0 | 85.3 | 84.7 |
| | GLM-5.1 | **90.0** | 89.3 | 88.7 | **90.0** | 89.3 | 88.7 | 88.0 | 87.3 | 86.7 |

Table 7: DeepConf voting accuracy (%) over the 3 reductions $\times$ 3 window sizes grid (generated tokens). The Lowest-Group $\times$ $W$=1024 cell matches the DC headline in Table 1. The highest value in each row is bolded.

| Token subset | $\lvert\text{copy}\rvert$ | $\lvert\overline{\text{copy}}\rvert$ | gap |
|---|---|---|---|
| all alnum (baseline) | 936 332 | 84 584 | $+0.51$ |
| stopword-removed | 558 491 | 74 714 | $+0.54$ |
| IDF-weighted | 558 491 | 74 714 | $+0.69$ |
| digits / cap-leading | 232 855 | 21 690 | $+0.44$ |
| length $\geq 8$ | 92 233 | 25 935 | $+0.62$ |

Table 8: Copy-vs-non-copy logprob gap under five token-subset rules. All gaps positive at $p \ll 0.001$ (paired bootstrap over rollouts).

```
When you have enough evidence, produce
a comprehensive markdown report
containing:
- A bolded direct answer to the
question
- ## Step-by-Step Reasoning and
Evidence section with sub-sections
walking through each clue
- A markdown table or bullet list
summarizing how each criterion is
satisfied
- Inline citations with [docid] for
every nontrivial claim
Do not give a one-line answer; produce
a structured research report.
```

### I.2 Force-final prompt

Issued once when the token-count guard fires, max-iterations is hit, the output budget is exhausted, or the API returns a non-retryable error. Tools are disabled for this call.

```
The conversation has been truncated.
Based on the evidence you have gathered
so far, write the final markdown report
now. Use the format from the system
prompt (bolded direct answer, ##
Step-by-Step Reasoning and Evidence, a
summary table, inline [docid]
citations). End the report with a
single line of the form Confidence:
NN%.
```

### I.3 Verbalised-confidence solicitation

Issued once after the rollout has produced a final answer (either naturally or via force-final), at temperature 0 with tools disabled.

```
Reply with ONLY a single integer
between 0 and 100 representing your
confidence (in percent) that the answer
you just gave is correct. No words, no
%, no labels — just the integer.
Example output: 73
```

### I.4 Judge prompt

The judge (Qwen3-32B at temperature 0) receives the question, the gold answer, and the rollout's final markdown report, and decides correctness. We use the official BrowseComp-Plus `GRADER_TEMPLATE` for every dataset.

```
Question: {question}
Gold answer: {gold_answer}
Model response: {model_response}

Read the model's response. First,
extract the model's final short answer
in the extracted_final_answer field (or
output None if the model did not
```

```
produce one). Then decide whether the
extracted answer matches the gold
answer up to obvious paraphrase /
equivalent name. Output a JSON object
with fields extracted_final_answer,
reasoning, and judge_correct ∈ { yes,
no }. Do not output anything else.
```

### I.5 Example thought span vs. answer prose

A single rollout on the BrowseComp-Plus question *"There is an influencer whose career has lasted over a decade... What is the full name of the influencer?"* (gold answer: *Tóke Makinwa*). The *thought span* is what DeepConf reads—chain-of-thought across all turns; the *answer prose* is what RGV reads—the assistant's output on the final answer turn (no internal thought).

**Thought span (excerpt, what DeepConf reads).**

```
Need an influencer with a >10-year
career who anchored a 2022 talk
segment. . . The host is born in the
1980s and has served as a host at major
events. Possible candidates: Tope
Mark-Odigie, Tóke Makinwa, Lilly
Singh. . . Tóke Makinwa is listed as 2nd
of 4 siblings on Wikipedia [docid 118],
became brand ambassador for Pulse
Nigeria in 2016. . . Lilly Singh is also
2nd of 4 but the 2022 talk-segment host
criterion fits Tóke better. . .
```

**Answer prose (what RGV reads).**

```
Tóke Makinwa.
## Step-by-Step Reasoning and Evidence
- Long-career influencer with a YouTube
vlog series: [docid 118] records that
Tóke Makinwa launched the Toke Moments
vlog in 2014, more than a decade after
her 2004 radio debut.
- 2022 talk-segment anchor: [docid 118]
notes she anchored the talk segment of
The Buzz, hosted by Ebuka Obi-Uchendu
(born 1982).
- Second of four siblings: [docid 118]
lists her as the second of four.
- 2016 ambassador for an MIT-founded
brand: [docid 118] records the Pulse
Nigeria ambassadorship; Pulse's founder
Tunde Kehinde is an MIT Sloan alumnus.
```

$\mathcal{T}(P_i)$ is built from the *Answer prose* block only; DeepConf's per-token confidences are computed over the *Thought span* only. The two operate on disjoint parts of the rollout's output.

### I.6 Example clustering on one question

For the same question, the eight rollouts produced the following predicted-answer strings, normalised under the strict rule:

| # | raw predicted answer | normalised | cluster |
|---|---|---|---|
| 1 | Tóke Makinwa. | toke makinwa | A |
| 2 | The answer is Toke Makinwa. | toke makinwa | A |
| 3 | Lilly Singh | lilly singh | B |
| 4 | Lilly Singh. | lilly singh | B |
| 5 | Lilly Singh | lilly singh | B |
| 6 | Toke Makinwa | toke makinwa | A |
| 7 | Shannon LaNier | shannon lanier | C |
| 8 | Bhuvan Bam | bhuvan bam | D |

Strict clustering recovers four clusters with $|A|$=3, $|B|$=3, $|C|$=$|D|$=1. Simple-majority ties A and B at weight 3; under DeepConf the eight rollouts all receive nearly identical weights and the tie persists; under RGV the three A-cluster rollouts carry the largest prose-recall (their answer prose lexically aligns with the *Tóke Makinwa* Wikipedia entry the agent actually fetched), so the weighted vote selects cluster A.

## J Success and failure cases

We first show three representative *successes*—minority-correct questions where RGV rescues the answer that DeepConf misses—then five *failures* exhibiting the "well-grounded but wrong" pattern of §6.4.

### J.1 Success cases: RGV rescues the correct minority

DeepConf gives all rollouts near-identical confidence and follows the wrong majority; RGV elevates the correctly grounded rollout.

**Success 1 (qid 301): royal family identification.** *Q:* "Name the royal family a certain individual's spouse was born into..." (multiple biographical constraints).
*Gold:* Nwoko.

| | | RGV | DC | Predicted answer |
|---|---|---|---|---|
| r2 | ✓ | **0.184** | 6.54 | Nwoko Royal Family |
| r6 | ✓ | 0.118 | 6.66 | Nwoko Royal Family |
| r7 | ✗ | 0.071 | 6.46 | Greek royal family |
| r1 | ✗ | 0.026 | 6.70 | Greek royal family |
| r5 | ✗ | 0.024 | 6.68 | British royal family |
| r0 | ✗ | 0.009 | 6.75 | House of Orange-Nassau |
| r4 | ✗ | 0.008 | 6.59 | House of Windsor |

Seven distinct answers. The two correct rollouts retrieve the Nwoko family page and echo it in their prose (RGV $\geq$ 0.118); all wrong rollouts name famous European royal families without retrieving relevant evidence (RGV $\leq$ 0.071). DeepConf range: 6.46–6.75, unable to discriminate.

**Success 2 (qid 424): architect identification.** *Q:* "Name the architect who was a WWII veteran and

TV broadcaster, designed a building completed 1977–1987…"
*Gold:* Raffaele Contigiani.

| | | RGV | DC | Predicted answer |
|---|---|---|---|---|
| r4 | ✓ | **0.152** | 6.98 | Raffaele Contigiani |
| r6 | ✓ | 0.133 | 6.72 | Raffaele Contigiani |
| r1 | ✗ | 0.095 | 6.75 | Victor Alfred Lundy |
| r3 | ✗ | 0.092 | 6.99 | Denys Lasdun |
| r2 | ✗ | 0.073 | 6.76 | Denys Lasdun |
| r0 | ✗ | 0.018 | 6.48 | Denys Lasdun |

Three wrong rollouts converge on Denys Lasdun (a famous British architect who fits some but not all constraints). The correct rollouts retrieve Contigiani's Italian biography and echo it in prose, producing a clear RGV separation (0.152 vs. 0.095).

**Success 3 (qid 56): movie identification.** *Q:* "Movie with a ReFrame Stamp, 2018–2023, about a festival, with a cast member who appeared in a heist film…"
*Gold:* Last Christmas.

| | | RGV | DC | Predicted answer |
|---|---|---|---|---|
| r2 | ✓ | **0.115** | 7.53 | Last Christmas |
| r5 | ✗ | 0.028 | 7.96 | Crazy Rich Asians |
| r4 | ✗ | 0.026 | 7.54 | Crazy Rich Asians |
| r3 | ✗ | 0.026 | 7.69 | Crazy Rich Asians |
| r7 | ✗ | 0.018 | 7.78 | Rifkin's Festival |
| r1 | ✗ | 0.018 | 7.37 | Rifkin's Festival |
| r6 | ✗ | 0.016 | 7.70 | A Simple Favor |

Three rollouts vote for Crazy Rich Asians with high DeepConf (7.54–7.96); DeepConf selects this wrong majority. RGV: the single correct rollout retrieves the film's page (0.115 vs. $\leq$0.028) and wins.

### J.2 Failure cases: well-grounded but wrong

**Failure 1 (qid 1185): cricket match.** *Q:* "Match number, tournament, year satisfying nine batting constraints…"
*Gold:* 31st match, IPL 2013.

| | | RGV | DC | Predicted answer |
|---|---|---|---|---|
| r7 | ✗ | **0.235** | 6.93 | 2019 ICC World Cup |
| r5 | ✗ | 0.174 | 6.98 | Bangladesh vs South Africa |
| r2 | ✓ | 0.141 | 7.30 | 31st match, IPL 2013 |

r7 retrieves extensive World Cup documentation and echoes cricket statistics (RGV = 0.235), outscoring the correct rollouts (0.165, 0.141) despite naming the wrong tournament.

**Failure 2 (qid 790): writer's work identification.** *Q:* "A writer born in July, article posted April 2023, smoked on Christmas…name the work."
*Gold:* "Manos".

| | | RGV | DC | Predicted answer |
|---|---|---|---|---|
| r0 | ✗ | **0.229** | 6.96 | "Killing Joke" |
| r3 | ✓ | 0.206 | 6.68 | "Manos" |
| r1 | ✗ | 0.094 | 6.71 | (different work) |
| r7 | ✗ | 0.008 | 6.70 | "mouse" |
| r4 | ✗ | 0.000 | 6.80 | "girls" |
| r5 | ✗ | 0.000 | 6.83 | "Thugs" |

r0 retrieves an article about the same writer but identifies a different work (0.229 vs. 0.206). Both rollouts ground their answer in the writer's bibliography; the wrong one happens to echo a more frequently cited title.

**Failure 3 (qid 773): clothing colour.** *Q:* "A child reported missing multiple times 2014–2018. What colour shirt in the 2018 police description?"
*Gold:* Red.

| | | RGV | DC | Predicted answer |
|---|---|---|---|---|
| r7 | ✗ | **0.209** | 7.41 | White t-shirt |
| r3 | ✗ | 0.190 | 7.27 | (different colour) |
| r4 | ✗ | 0.180 | 7.68 | (different colour) |
| r2 | ✗ | 0.179 | 7.46 | (different colour) |
| r5 | ✗ | 0.166 | 7.88 | (different colour) |
| r6 | ✗ | 0.142 | 8.05 | (different colour) |
| r1 | ✓ | 0.109 | 7.11 | Red |
| r0 | ✗ | 0.000 | 6.45 | (different colour) |

All seven wrong rollouts retrieve articles about the same missing-person case and echo detailed police descriptions, receiving higher RGV scores (0.142–0.209) than the single correct rollout (0.109). The correct answer ("Red") is a single word with minimal lexical overlap; wrong rollouts produce longer prose quoting extensively from the retrieved reports.

**Pattern and future directions.** Each failure shares a structure: the wrong rollout retrieves documents about the right *domain* (same writer, same sport, same missing-person case) but identifies the wrong *specific entity* within that domain. RGV measures whether the answer is anchored in retrieved evidence; it cannot verify that the evidence answers the specific question asked. Closing this gap likely requires moving beyond surface-level lexical overlap toward *semantic* grounding checks, for example, entailment verification between the question constraints and the retrieved passages, or cross-referencing the answer against multiple independent sources. Such extensions could complement RGV's efficiency with deeper reasoning at a modest additional cost.