When Agents Disagree With Themselves: Behavioral Consistency as an Uncertainty Signal for LLM Agents

Aman Mehta [1]

Abstract

Running the same LLM agent on identical inputs yields 2.3–4.2 distinct action sequences per 10 runs; this behavioral variance constitutes a training-free, black-box uncertainty signal that instantiates selective classification and distribution-free calibration for agentic systems. Across 8,000 runs of four models on 200 HotpotQA questions, consistent tasks ($\leq$2 unique paths) achieve 82–87% accuracy while inconsistent tasks ($\geq$4 paths) achieve 41–65%, a gap that survives controls for task difficulty. Divergence concentrates at step 2 (50.5% of Llama tasks), and consistency metrics detect failures with AUROC 0.62–0.78. Exploiting this signal, selective prediction (answering only when $k = 3$ runs agree) achieves 87–88% accuracy at 54–62% coverage, a 6–14pp gain over single-run baselines, and matches a split-conformal baseline without a held-out calibration set. A cross-benchmark validation on SWE-bench (50 tasks, 1,000 runs) preserves the consistency hierarchy while revealing an $\sim$8$\times$ spread in mean trajectory length across models, and bootstrap analysis shows single-run evaluations misrank models 29.3% of the time.

1. Introduction

Large language model (LLM) based agents that use tools and multi-step reasoning are increasingly deployed for complex tasks (Yao et al., 2022; Schick et al., 2023). As these systems move from prototypes to production, understanding their reliability becomes critical.

A fundamental but understudied question is: *how consistent are LLM agents in their behavior?* Given identical inputs, will an agent follow the same reasoning path and arrive at the same answer? This matters because inconsistency complicates debugging, may signal uncertainty exploitable for error detection, and could inform architecture decisions.

We connect behavioral consistency to the statistical frameworks of uncertainty estimation and selective prediction. Each independent agent run can be viewed as an implicit ensemble member (Lakshminarayanan et al., 2017), and disagreement across runs constitutes a model-agnostic uncertainty signal. Unlike verbalized confidence (Xiong et al., 2024) or learned calibration (Kadavath et al., 2022), this signal requires no model modification, no access to internal states, and no training; it emerges naturally from repeated execution. We show this signal is actionable: it enables selective prediction (abstaining on uncertain inputs), runtime failure detection, and more reliable benchmark evaluation.

We present a systematic empirical study of behavioral consistency in ReAct-style agents (Yao et al., 2022), instantiating three statistical frameworks central to the agentic uncertainty agenda (selective classification, distribution-free calibration, and sequential monitoring) through a single training-free, black-box signal: cross-run agreement. Our key contributions:

1. **Filter $>$ aggregator asymmetry (new for agents).** In multi-step agents, majority voting yields only +0–2pp (vs. +5–17pp for single-turn CoT (Wang et al., 2023)) because errors are systematic; *filtering* on disagreement instead yields +6–14pp for 3 of 4 models. We explain the gap via a 2$\times$2 task taxonomy in which *consistent-wrong* tasks (5.5–10%) set the hard ceiling on filtering (Section 4.4).
2. **Selective prediction with distribution-free validation.** We instantiate the Geifman–El-Yaniv selective-classification framework for agents using cross-run agreement as the selection function, prove a Hoeffding-style concentration bound on the agreement score (Prop. 1), and show empirically that the resulting risk-coverage curve matches a split-conformal baseline (Angelopoulos & Bates, 2023) *without* a held-out calibration set, achieving 87–88% accuracy at 54–62% coverage (Sections 3.4, 4.6).
3. **Behavioral $>$ verbalized confidence.** On a head-to-head comparison on a common 100-question split,

[1]Snowflake AI Research. Correspondence to: Aman Mehta <aman.mehta@snowflake.com>.

Accepted to the ICML 2026 Workshop on Statistical Frameworks for Uncertainty in Agentic Systems, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

behavioral agreement outperforms verbalized self-confidence as a failure detector across three models (Claude, GPT-5, Llama; AUROC 0.65–0.74 vs. 0.48–0.55), combining the two does not improve over behavioral alone, and behavioral consistency requires no prompt engineering, no log-probs, and no model modification (Section 4.3).

4. **Cross-benchmark stability.** The consistency hierarchy (Claude < GPT-5 < Gemini < Llama by CV) is preserved exactly across HotpotQA (3-tool QA, 200 tasks) and SWE-bench Verified (unrestricted bash, 50 tasks across 5 repos, 1,000 runs), suggesting consistency is a stable, benchmark-independent model property (Section 4.7).

5. **Ranking instability and divergence mechanism.** Single-run evaluations misrank models 29.3% [28.4–30.1%] of the time, undermining standard benchmark methodology (Section 4.8); divergence concentrates at step 2 (50.5% for Llama vs. 4–6% for Claude/GPT-5), and step-level agreement constitutes a running test statistic for sequential monitoring (Ramdas et al., 2023) (Section 4.5).

2. Related Work

LLM-Based Agents. ReAct (Yao et al., 2022) introduced interleaving reasoning with actions. Subsequent work extended this to web browsing (Zhou et al., 2023), code execution (Jimenez et al., 2023), and multi-agent systems (Wu et al., 2023). We study reliability under repeated execution.

Agent Consistency and Reliability. τ-bench (Yao et al., 2024) showed that GPT-4o's pass rate drops from 60% (pass^1) to 25% (pass^8). While τ-bench quantifies *whether* agents are inconsistent, we investigate *where* and *why*: tracing divergence to step 2, correlating consistency with correctness, and identifying path length as a signal. Kapoor et al. (2024) argue that benchmarks overestimate agent capabilities; our findings support this. Concurrent work has begun formalizing the reliability landscape for agents. ReliabilityBench (Gupta, 2026) defines a unified reliability surface across consistency, robustness, and fault tolerance; Rabanser et al. (2026) propose twelve metrics decomposing reliability along four axes. We provide the deepest empirical analysis of the consistency dimension specifically, demonstrating that it functions not just as a reliability metric but as an actionable uncertainty signal for selective prediction and failure detection.

Agent Uncertainty Quantification. Oh et al. (2026) present the first general formulation of agent UQ and identify the lack of fine-grained multi-run benchmarks as a key challenge; our 8,000-run dataset and consistency analysis contribute directly to this agenda. Zhang et al. (2026) propose a training-based dual-process framework for propagating verbalized uncertainty through agent trajectories; behavioral consistency provides a complementary training-free, black-box signal that requires no model modification.

Uncertainty and Selective Prediction. Our work connects to three threads. First, *ensemble disagreement*: deep ensembles use prediction variance as an uncertainty estimate (Lakshminarayanan et al., 2017); behavioral consistency across agent runs is an analogous signal, where each run acts as an implicit ensemble member. Recent work has applied conformal prediction to LLM outputs (Kumar et al., 2023; Quach et al., 2024) and to robot planning under uncertainty (Ren et al., 2023), providing distribution-free coverage guarantees. Farquhar et al. (2024) extend semantic uncertainty to detect LLM hallucinations via predictive entropy, requiring access to token-level log-probabilities. Behavioral consistency provides a complementary signal at the trajectory level: it captures uncertainty in the agent's *plan*, not just its output tokens, and operates as a black-box method requiring only repeated execution. Crucially, two runs can produce the same final answer via divergent tool sequences, a source of plan-level uncertainty that token-level entropy cannot detect. Second, *self-consistency*: Wang et al. (2023) show that majority voting over sampled reasoning chains improves accuracy; we extend this from single-turn CoT to multi-step agentic settings and additionally study consistency as a *diagnostic* signal, not just an ensembling strategy (Section 4.4). Third, *selective prediction*: Geifman & El-Yaniv (2017) and Kamath et al. (2020) show that abstaining on uncertain inputs improves precision; our consistency-filtered results instantiate this for agents. Unlike verbalized confidence (Tian et al., 2023; Xiong et al., 2024), behavioral consistency requires no model modification; it emerges naturally from repeated execution. Prior work on LLM calibration (Kadavath et al., 2022) and semantic uncertainty (Kuhn et al., 2023) focuses on single-turn outputs; we extend to multi-step agentic behavior.

3. Methodology

3.1. Agent Architecture

We implement a ReAct-style agent with three tools: **Search(query)** returns document titles via keyword matching, **Retrieve(title)** returns full text, and **Finish(answer)** terminates with a final answer. The agent follows the standard think-act-observe loop.

3.2. Experimental Setup

Dataset. We use 200 questions from HotpotQA (Yang et al., 2018) validation (distractor setting), requiring multi-hop reasoning over 10 paragraphs (2 gold, 8 distractors).

Models. Four models across four providers: Llama 3.1 70B (Meta), GPT-5 (OpenAI), Claude Sonnet 4.5 (Anthropic), and Gemini 3 Pro (Google). The per-run figures (Sections 4.5, 4.6, Figures 5–6) and the verbalized-confidence comparison (Section 4.3, Table 4) use 100-question splits with full per-run trajectories ($k = 10$ for the figures, $k = 5$ for Table 4). As an earlier-generation reference, we additionally evaluated GPT-4o on a separate 100-question subset (Appendix E).

Runs. For each question–model pair, 10 independent runs at temperature 0.7, yielding 8,000 total runs. Gemini 3 Pro required an additional `top_p=0.95` parameter to achieve stochastic sampling (Appendix A). Temperature ablation: 20 questions × 5 runs × 4 models × 3 temperatures.

Statistical note. All tests are two-sided; raw p-values without multiple-comparison correction given the exploratory nature.

3.3. Metrics

Answer consistency: fraction of runs producing the most common answer. **Action sequence diversity**: number of unique action sequences across N runs. **Step variance ratio**: $(\max - \min)/\text{mean}$ of step counts. **First divergence point**: earliest step where runs take different actions. **Correctness**: fuzzy string matching (answer contains gold or vice versa, case-insensitive); results under exact match and token F1 in Appendix G.

3.4. Behavioral Consistency as Selective Prediction

We formalize behavioral consistency within the selective classification framework (Geifman & El-Yaniv, 2017). Let f be an agent, x a task, and $\{f_1(x), \ldots, f_k(x)\}$ be k independent runs at fixed temperature. Define the *agreement score*:

$$s(x) = \max_a \frac{1}{k} \sum_{i=1}^{k} \mathbf{1}[f_i(x) = a] \tag{1}$$

i.e., the fraction of runs producing the plurality answer. The *selection function* $g_\tau(x) = \mathbf{1}[s(x) \geq \tau]$ admits task x when agreement exceeds threshold τ. The *selective risk* at threshold τ is:

$$R(\tau) = \frac{\sum_i \ell(f(x_i), y_i) \cdot g_\tau(x_i)}{\sum_i g_\tau(x_i)} \tag{2}$$

where ℓ is 0-1 loss and $f(x_i)$ is the plurality answer.

Proposition 1 (Concentration of agreement score)**.** *Let $s^*(x) = \max_a \Pr[f(x) = a]$ be the true plurality probability for task x, and $\hat{s}_k(x)$ the empirical agreement score from k i.i.d. runs (Eq. 1). Then for any $\delta > 0$:*

$$\Pr\left[|\hat{s}_k(x) - s^*(x)| \leq \sqrt{\tfrac{\ln(2/\delta)}{2k}}\,\right] \geq 1 - \delta \tag{3}$$

Table 1. Performance across four models (200 tasks each). Seqs = unique action sequences; Var = step variance ratio.

Model	Correct	Seqs	Steps	Var
Claude Sonnet 4.5	**81.5%**	**2.3**	5.1	**22.2%**
GPT-5	79.6%	2.4	4.2	36.0%
Llama 3.1 70B	73.7%	4.2	4.7	64.9%
Gemini 3 Pro	72.2%	3.2	5.4	38.5%

If $\hat{s}_k(x) \geq \tau$ and $\epsilon_k = \sqrt{\ln(2/\delta)/(2k)}$, the true plurality probability satisfies $s^*(x) \geq \tau - \epsilon_k$ with probability $\geq 1 - \delta$.

Proof. Hoeffding's inequality applied to k i.i.d. Bernoulli indicators $\mathbf{1}[f_i(x) = a^*]$. □

For $k = 3$, $\epsilon_3 = 0.78$ at $\delta = 0.05$ (vacuous); for $k = 10$, $\epsilon_{10} = 0.43$; the bound becomes informative at $k \geq 20$ ($\epsilon_{20} = 0.30$). Despite the looseness at small k, our empirical risk-coverage curves (Figure 4) confirm that $R(\tau)$ decreases monotonically with τ at $k = 10$, and Section 4.6 validates the agreement score against a split-conformal baseline.

Exchangeability assumption. Eq. 3 requires $\{f_i(x)\}_{i=1}^k$ to be i.i.d. given x. Our setup satisfies this: each run uses an independent API call with a fresh random seed and stateless tools (search and retrieve operate over a fixed corpus). Exchangeability would be violated by stateful tools (e.g., a shared database modified by earlier runs) or autoregressive dependence across runs. While we cannot rule out provider-side prompt caching, such caching affects only latency, not output distributions, under standard API contracts. Our design avoids all substantive violations.

4. Results

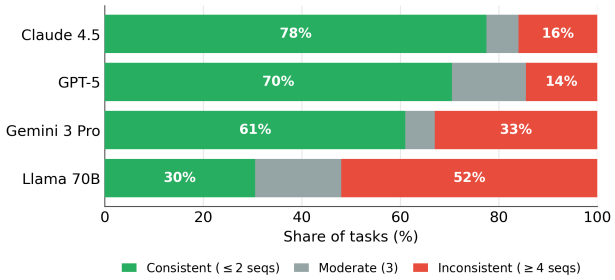
4.1. Overall Model Comparison

Table 1 summarizes 8,000 runs. Claude Sonnet 4.5 achieves both the highest accuracy and consistency (2.3 unique sequences). Llama 3.1 70B shows the most behavioral variance (4.2 unique sequences), followed by Gemini 3 Pro (3.2). All four models exhibit measurable behavioral inconsistency even on identical inputs.

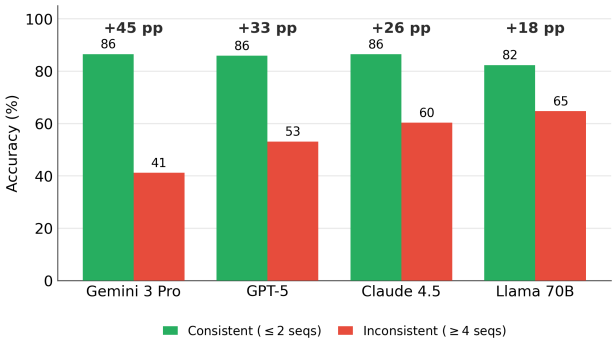
4.2. Consistency Is Associated with Correctness

Table 2 shows the central finding: consistent tasks (≤2 unique sequences) achieve 82–87% accuracy while inconsistent tasks (≥4 sequences) achieve 41–65%. The gap is significant for all four models (Mann-Whitney U, all $p < 0.001$; rank-biserial $r = -0.35$ to -0.61).[1]

[1] All four consistency-gap tests remain significant at $\alpha = 0.05$ after Holm-Bonferroni correction (most stringent adjusted $p = 3.44 \times 10^{-5}$).



(a) Share of tasks by behavioral consistency level (unique action sequences).



(b) Correctness for consistent ($\leq$2 seqs) vs. inconsistent ($\geq$4 seqs) tasks.

Figure 1. Behavioral consistency varies across models (left) and is associated with correctness (right).

Table 2. Correctness (%) by consistency level. Gap in pp. r = rank-biserial; n = consistent/inconsistent counts. All $p < 0.001$.

Model	Cons. ($\leq$2)	Incons. ($\geq$4)	Gap	r	n
Gemini 3 Pro	86.5	41.2	45.3	−.61	122/66
GPT-5	85.9	53.1	32.8	−.43	141/29
Claude Son. 4.5	86.4	60.3	26.1	−.39	155/32
Llama 70B	82.3	64.7	17.6	−.35	61/104

Table 3. Failure detection via consistency metrics. H = answer entropy. Baseline precision equals the per-model error rate.

Model	Feat.	Prec.	Rec.	AUROC
Gemini 3 Pro	Seq>4	72.1%	54.4%	.775
Llama 70B	H>1.5	44.9%	50.0%	.714
GPT-5	H>1.5	35.5%	56.4%	.688
Claude Son. 4.5	H>0.5	23.9%	63.6%	.619

Controlling for difficulty. Task difficulty confounds the consistency–correctness relationship. We compute a difficulty proxy (mean correctness across all models) and bin questions into Easy ($\geq$0.80, $n = 132$), Medium (0.40–0.79, $n = 32$), and Hard ($<$0.40, $n = 36$). Within Easy tasks, the consistency gap is positive for Gemini (+16.5pp), Llama (+11.2pp), and GPT-5 (+6.8pp) but near zero for Claude (−0.5pp), where ceiling effects leave little room for a gap (both consistent and inconsistent Easy tasks exceed 90% accuracy). Within Medium tasks, the gap is positive for all four models (+7 to +34pp), though cell sizes are small (n_c=6–16, n_i=10–22; Appendix C). For Hard tasks, the gap reverses for three of four models under fuzzy match. This reversal is a *consistent-wrong* phenomenon: of 61 consistent Hard task-model pairs, 52 (85.2%) are scored incorrect under fuzzy match. Approximately 30% of these are false negatives, surface-form mismatches where the model produced a semantically correct answer (e.g., "Pasek and Paul" vs. gold "Pasek & Paul"; "Kelly Osbourne" vs. "Kelly Lee Osbourne"). Annotation protocol: one annotator classified each case as surface-form mismatch vs. genuinely incorrect; all annotated cases and classification criteria will be released. When we recompute Hard stratum gaps under token F1 $\geq$ 0.5, all four models show non-negative gaps (+9 to +16pp; Table 10), confirming that the reversal is a fuzzy-match artifact. The remaining $\sim$70% represent true consistent-wrong behavior, where the agent confidently commits to a wrong answer. Full analysis in Appendix C. The within-stratum association is weaker than the aggregate effect, consistent with difficulty being the primary confound. Partial correlations (controlling for difficulty as a continuous variable) remain significant for three of four models: Gemini ($r = -0.49$, $p < 0.001$), GPT-5 ($r = -0.16$, $p = 0.025$), and Llama ($r = -0.16$, $p = 0.028$). The non-significant partial correlation for Claude ($r = 0.01$, $p = 0.88$) reflects a ceiling effect: Claude is consistent-correct on 68% of tasks, leaving insufficient within-model variance for the partial correlation to detect.

4.3. Consistency as a Runtime Failure Detector

We frame failure detection as binary classification: predicting whether the majority answer is incorrect using only consistency metrics. Table 3 reports the best feature per model.

For the most variable models (Llama, GPT-5, Gemini), consistency metrics achieve AUROC 0.69–0.78, substantially above the 0.50 random baseline. Even for the most consistent model (Claude), AUROC reaches 0.62, validating consistency as a lightweight runtime failure signal across all four models. Gemini's best predictor is unique sequences rather than answer entropy (>4 sequences indicates likely failure); with 66 inconsistent tasks in 200 questions, Gemini provides sufficient signal for the sequence-count feature to outperform entropy-based thresholds.

Table 4. Failure detection AUROC: behavioral consistency vs. verbalized confidence vs. combined (logistic regression, 5-fold CV), on a common 100-question split ($\times$ 5 runs). Behavioral consistency outperforms verbalized confidence for all three models, and combining the two does not improve over behavioral consistency alone.

Model	Behavioral	Verbalized	Combined
Claude Son. 4.5	.651	.522	.646
GPT-5	.734	.547	.727
Llama 70B	.739	.481	.701

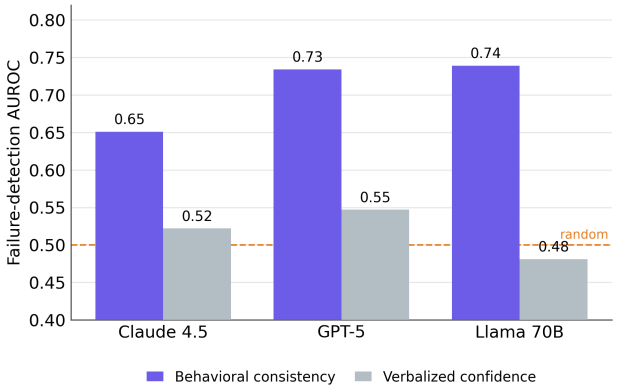


Figure 2. Behavioral consistency outperforms verbalized self-confidence as a failure detector for all three models, with verbalized confidence near or below the random baseline.

Comparison to verbalized confidence. Table 4 compares behavioral consistency to verbalized confidence (post-hoc 0–100 self-rating elicited via a single additional API call after each run) as failure detectors.[2] For the three models on a common 100-question split (Claude, GPT-5, Llama), behavioral consistency outperforms verbalized confidence (Claude 0.651 vs. 0.522; GPT-5 0.734 vs. 0.547; Llama 0.739 vs. 0.481; Figure 2). Spearman correlation between the two signals is low (largely orthogonal), and combining them via logistic regression (5-fold CV) does not improve over behavioral consistency alone, suggesting behavioral consistency subsumes the predictive information in verbalized confidence. These results are consistent with prior findings that behavioral consistency is a competitive, training-free failure signal requiring no prompt engineering, no log-probs, and no model modification.

4.4. Majority Voting and Selective Prediction

Can consistency be *exploited*, not just observed? We simulate a majority-vote intervention using existing multi-run data: for budget k, subsample k runs per question, take the majority answer, and evaluate accuracy (500 bootstrap iterations).

[2]Behavioral AUROCs in Table 4 differ from Table 3 because they are computed on a 100-task subset with $k=5$ runs rather than the full 200 tasks with $k=10$.

Table 5. Majority-vote accuracy (%) and selective prediction. $k=1$ is single-run baseline. Sel. = unanimous $k=3$ (answer only when all 3 agree).

Model	$k=1$	$k=3$	Gain	Sel. Acc	Cov.
Llama 70B	73.6	75.6	+2.0	87.2	53.8
Claude Son. 4.5	81.5	82.2	+0.7	87.8	61.8
GPT-5	79.6	80.1	+0.5	88.1	55.2
Gemini 3 Pro	72.2	72.3	+0.1	73.4	64.7

Table 6. Task-outcome taxonomy (all 200 tasks; per-model ranges). Filtering retains the top-left cell and excludes the bottom-right; Consistent-Wrong tasks (top-right) set the performance ceiling.

	Majority Correct	Majority Wrong
Consistent ($\leq$2 seqs)	Cons.-Correct 25–68%	**Cons.-Wrong** 5.5–10%
Inconsistent ($\geq$4 seqs)	Incons.-Lucky 8–38%	Incons.-Wrong 6–19%

We instantiate the selective classification framework (Geifman & El-Yaniv, 2017) for multi-step agents, using behavioral agreement as the selection function g_τ (Eq. 1). Table 5 reveals that consistency works far better as a *filter* than an *aggregator*, a striking contrast with single-turn self-consistency (Wang et al., 2023), where majority voting improves CoT accuracy by 5–17pp. In our multi-step agentic setting, voting yields only +0–2pp because agentic errors are *systematic*: an early wrong action commits the agent to an incorrect trajectory, producing consistent-wrong answers that voting cannot correct. *Selective prediction* (answering only when all $k = 3$ runs agree) is the stronger intervention: for Llama, Claude, and GPT-5, unanimous agreement achieves 87–88% accuracy (6–14pp above baselines) at 54–62% coverage. Figure 3 visualizes this asymmetry directly.

Table 6 gives the 2×2 task-outcome taxonomy: consistent-correct tasks (25–68%) are retained; inconsistent-wrong tasks (6–19%) are correctly excluded; *consistent-wrong* tasks (5.5–10%) set the hard ceiling on filtering gains.

Notably, 61% of Claude's errors are consistent-wrong vs. only 25% of Llama's, explaining why Llama benefits most from voting. Stricter thresholds yield further gains: unanimous 5/5 agreement achieves 90.2% accuracy for GPT-5 at 48% coverage (full taxonomy and thresholds in Appendix H).

The computational cost of $k=3$ runs is $3\times$ the single-run cost. At current API pricing, $k = 3$ HotpotQA runs cost approximately \$0.03–0.15 per task depending on the model. The accuracy gain (6–14pp) and the ability to abstain on uncertain inputs justify this overhead in deployments where reliability matters more than throughput.

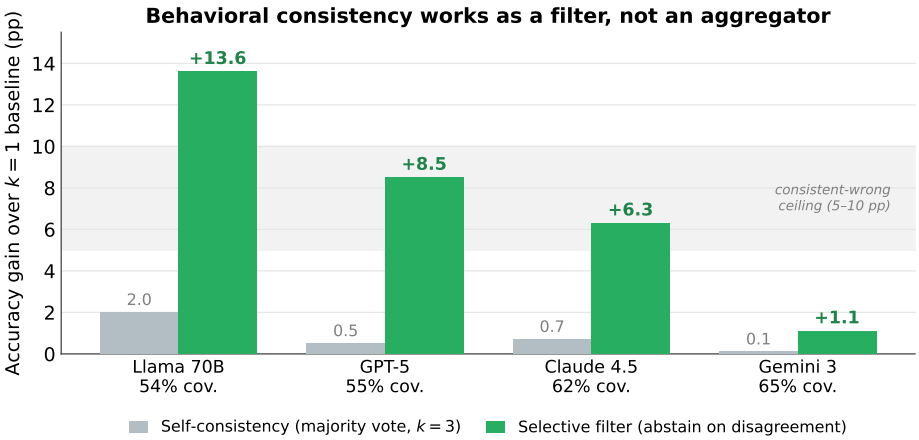


Figure 3. Filter > aggregator. Self-consistency (majority vote, $k = 3$) gains only 0–2pp, whereas behavioral filtering (answer only when all $k = 3$ runs agree) gains 6–14pp for three of four models at 54–62% coverage. The shaded band marks the consistent-wrong ceiling (5.5–10% of tasks), where the agent confidently commits to an incorrect answer, the fundamental limit on filtering gains.

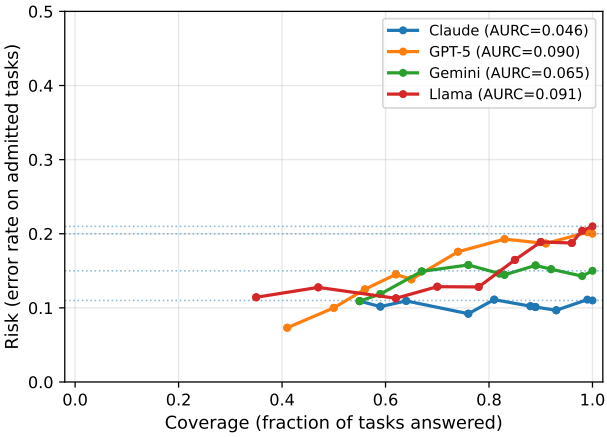


Figure 4. Risk-coverage curve. As the agreement threshold τ increases (moving left), admitted tasks have lower error rates. The monotonic decrease confirms that behavioral agreement is a valid selection function in the Geifman–El-Yaniv framework (Geifman & El-Yaniv, 2017). AURC (Zhou et al., 2025) values per model in legend.

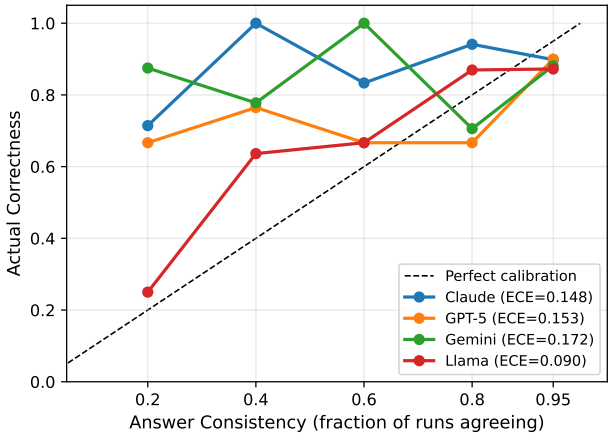


Figure 5. Behavioral calibration curve. Answer consistency (fraction of runs agreeing) vs. actual correctness. Dashed diagonal = perfect calibration. ECE values per model in legend.

Figure 4 plots the risk-coverage tradeoff as the agreement threshold τ varies. The monotonically decreasing curves confirm that behavioral agreement is a valid selection function: stricter agreement requirements yield lower risk at the cost of reduced coverage, consistent with Eq. 2.

Figure 5 shows the relationship between answer consistency and actual correctness. Behavioral consistency is an imperfect probability estimate: agreement levels do not match correctness exactly (ECE = 0.148, 0.153, 0.172, 0.090 for Claude, GPT-5, Gemini, Llama respectively). This calibration gap, analogous to miscalibration in neural network confidence (Guo et al., 2017), suggests that multi-run consistency is a useful but imperfect uncertainty signal. Importantly, this calibration property emerges without any training or model modification.

4.5. Divergence Occurs Early; Path Length Matters

Pooled across four models, 18.8% [95% CI: 16.2–21.6%] of tasks diverge by step 2, but this rate varies dramatically: Llama diverges early in 50.5% [43.6–57.4%] of tasks vs. 4–6% for Claude and GPT-5. Among Llama's early-diverging tasks, accuracy is 71.7% vs. 85.8% for late-diverging tasks. The first search query largely determines the trajectory.

Path length is strongly associated with behavioral consistency: Spearman ρ between mean steps per task and unique action sequences is 0.74 (Llama), 0.54 (Gemini), 0.53 (Claude), 0.50 (GPT-5); pooled $\rho = 0.51$, all $p < 0.001$. Path length also tracks correctness in the early-vs-late di-

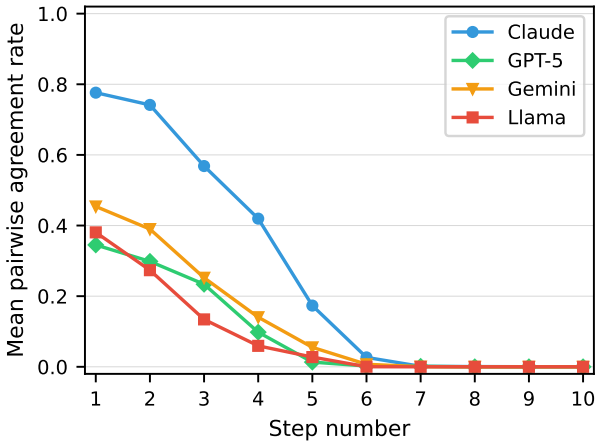


Figure 6. Mean pairwise action-sequence agreement rate per step (100 tasks × 10 runs, four models). Llama diverges earliest and most sharply; Claude retains high agreement through step 2. Step-2 agreement alone yields AUROC 0.49–0.60 for failure prediction.

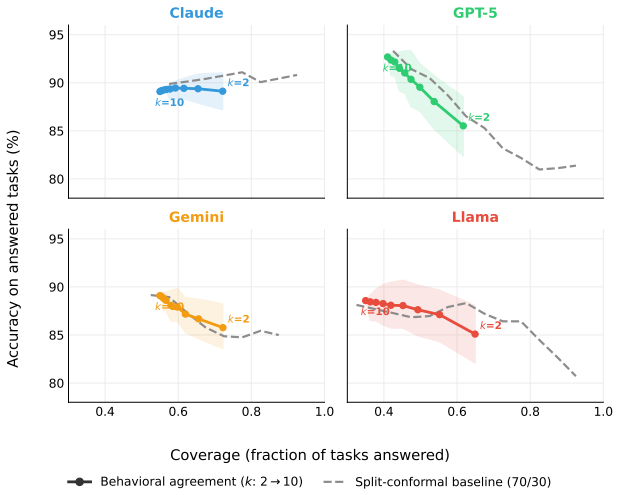


Figure 7. Coverage vs. accuracy as unanimity requirement k increases from 2 to 10. Shaded bands = 10–90% bootstrap intervals. Dashed: split-conformal baseline (70/30 calibration split). Behavioral consistency matches the conformal baseline without requiring a calibration set.

vergence comparison above (e.g., Llama 71.7% accuracy on early-diverging tasks vs. 85.8% on late-diverging tasks). Longer paths indicate backtracking and uncertainty; each additional step is an opportunity to diverge and err.

Figure 6 quantifies this timing by measuring the mean pairwise action-sequence agreement rate at each step across 100 tasks × 10 runs for all four models. Agreement drops after step 1 for all models; Llama exhibits the steepest decline (0.38 at step 1, 0.27 at step 2, 0.13 at step 3), while Claude retains high agreement through step 2 (0.74), consistent with its lower behavioral variance. Step-2 action agreement predicts final majority-vote correctness with AUROC 0.60 (Claude), 0.58 (Gemini), 0.50 (GPT-5), 0.49 (Llama), near chance, indicating that *where* divergence first occurs is informative about model behavior but that *answer-level* consistency (Table 3, AUROC 0.62–0.78) is the stronger failure signal.

4.6. Comparison to Split-Conformal Selective Prediction

A natural question is whether the agreement score provides genuine selective-prediction value, or whether a standard conformal procedure would achieve comparable results. We compare against a *split-conformal baseline*: calibrate the agreement threshold on 70% of tasks and evaluate on the held-out 30%, repeating over 500 random splits. This provides a distribution-free coverage guarantee at any target level (Angelopoulos & Bates, 2023; Bates et al., 2021).

Figure 7 plots coverage vs. accuracy as the unanimity requirement k increases from 2 to 10 for all four models (100 tasks × 10 runs each). Behavioral consistency closely tracks the conformal baseline for all four models, confirming that the agreement score is a well-calibrated selection function *without requiring a separate calibration set*. This is the key practical advantage: conformal methods need a held-out calibration split, while behavioral consistency uses only the k runs already executed for prediction, so no data is "wasted" on calibration.

Table 7. SWE-bench cross-benchmark validation (50 tasks × 5 runs across 5 repositories). CV = std/mean of step counts. The HotpotQA consistency hierarchy is perfectly preserved across all four models.

	Claude	GPT-5	Gemini	Llama
Tasks	50	50	50	50
Repositories	5	5	5	5
CV	**17.7%**	30.3%	46.8%	68.9%
Mean steps	53.3	10.5	81.8	27.9

4.7. Cross-Benchmark Validation: SWE-bench

To test generalizability beyond HotpotQA's minimal action space, we conduct a cross-benchmark validation on SWE-bench Verified (Jimenez et al., 2023; OpenAI, 2024): 50 tasks across 5 open-source repositories (astropy, django, matplotlib, scikit-learn, sympy), 5 runs per model-task pair (1,000 total runs), using all four models with an identical bash-only scaffold (temperature 0.5).

The consistency hierarchy is perfectly preserved across all four models (Claude < GPT-5 < Gemini < Llama by CV), matching the HotpotQA ranking. This hierarchy holds across a 3-tool QA environment (HotpotQA) and unrestricted-bash coding tasks spanning 5 open-source repositories (SWE-bench), suggesting that behavioral con-

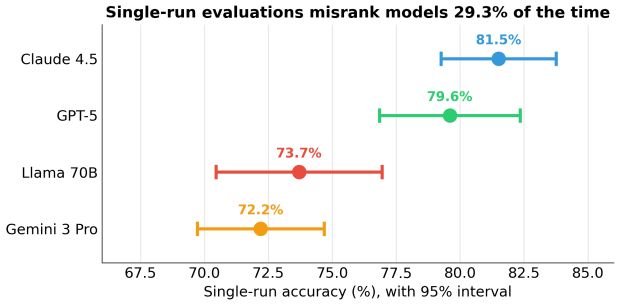


Figure 8. Single-run accuracy with 95% intervals (one run sampled per question). Adjacent models' intervals overlap, with Claude/GPT-5 near the top and Llama/Gemini at the bottom (1.5pp apart), so single-run evaluations reorder models 29.3% of the time.

sistency is a stable, benchmark-independent model property. Resolution rates on the original 10-task astropy subset were Claude 58%, GPT-5 32%, Llama 4%, preserving the accuracy hierarchy. Full cross-benchmark comparison in Appendix D.

Trajectory length spans ∼8×. Mean steps range from GPT-5 (10.5) to Gemini (81.8), with Llama (27.9) and Claude (53.3) in between, a 7.8× ratio the 3-tool HotpotQA setup masks (4.2–5.4; Table 1). Length and CV are *not* co-monotone (Gemini is the longest yet second-most-consistent; Llama is shorter than Claude/Gemini yet most variable), so they capture distinct dimensions, with direct cost-, latency-, and step-cap implications.

4.8. Ranking Instability

We perform 10,000 bootstrap iterations across four models on all 200 questions, sampling one run per question per model. **29.3% [28.4–30.1%] of single-run evaluations produce a ranking differing from the multi-run ground truth** (Claude 81.5% > GPT-5 79.6% > Llama 73.7% > Gemini 72.2%; Figure 8). Nearly one in three single-run evaluations would misrank models. The instability stems from narrow accuracy gaps between adjacent models and within-model variance. Agent benchmarks should report multi-run statistics.

5. Discussion

Practical implications. Consistency monitoring is feasible and actionable: even $k = 3$ runs deliver the selective-prediction gains of Section 4.4, the ceiling is set by the consistent-wrong fraction (5.5–10%; Table 12), and the step 2 bottleneck (Section 4.5) suggests that improving first-query formulation could reduce downstream variance.

Consistency as implicit, distribution-free calibration. Traditional calibration asks whether a model's *stated* confidence matches its accuracy (Kadavath et al., 2022); behavioral consistency offers an alternative *behavioral* notion in which cross-run disagreement plays the role of deep-ensemble variance (Lakshminarayanan et al., 2017) and, crucially, requires no calibration set, no log-probs, and no prompting strategy. This positions it alongside conformal prediction (Angelopoulos & Bates, 2023) and distribution-free risk control (Bates et al., 2021) as a fully black-box, training-free path to uncertainty in agentic systems.

Limitations and future work. Our analysis primarily uses HotpotQA with a 3-tool action space; the SWE-bench cross-benchmark validation provides evidence of generalizability across action spaces and repositories. The temperature ablation uses a smaller sample (20 questions). We cannot establish causality from observational data. Gemini 3 Pro required a `top_p` workaround to achieve stochastic sampling (Appendix A). The consistent-wrong annotation for Hard-stratum tasks was performed by a single annotator; all cases and classification criteria are provided for reproducibility. SWE-bench resolution rates are available only for the original 10-task astropy subset; the 50-task expanded validation uses behavioral metrics (CV, step counts) without per-task resolution data. We plan extensions to multimodal agents and analytical reasoning (FinQA, GAIA).

6. Conclusion

Behavioral consistency in LLM agents is measurable, associated with correctness (effect sizes that survive difficulty controls), and practically exploitable as both a runtime failure signal (AUROC 0.62–0.78) and a selective-prediction filter (87–88% accuracy at 54–62% coverage). With single-run evaluations misranking models 29.3% of the time, these findings argue for multi-run evaluation, consistency-based selective prediction, and runtime consistency monitoring as standard practice. Code and data will be released upon publication.

Impact Statement

We demonstrate that behavioral consistency provides a training-free, model-agnostic uncertainty signal for LLM agents. Our selective prediction results (87–88% accuracy at 54–62% coverage) and failure detection (AUROC 0.62–0.78) suggest practical deployment value for monitoring and routing in agentic systems. Our finding that single-run evaluations produce incorrect model rankings 29% of the time has implications for benchmark reliability. One risk of consistency-based filtering is that it may systematically abstain on underrepresented task subgroups (e.g., rare question types or minority-language inputs), masking failures that disproportionately affect certain users; deployments should pair behavioral filtering with disaggregated evalua-

tion across task demographics.

References

Angelopoulos, A. N. and Bates, S. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 2023.

Bates, S., Angelopoulos, A., Lei, L., Malik, J., and Jordan, M. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 2021.

Farquhar, S., Kossen, J., Kuhn, L., and Gal, Y. Detecting hallucinations in large language models using semantic entropy. *Nature*, 2024.

Geifman, Y. and El-Yaniv, R. Selective classification for deep neural networks. *NeurIPS*, 2017.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. *ICML*, 2017.

Gupta, A. ReliabilityBench: Evaluating LLM agent reliability under production-like stress conditions. *arXiv preprint arXiv:2601.06112*, 2026.

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. SWE-bench: Can language models resolve real-world GitHub issues? *arXiv preprint arXiv:2310.06770*, 2023.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Kamath, A., Jia, R., and Liang, P. Selective question answering under domain shift. *ACL*, 2020.

Kapoor, S., Stroebl, B., Siegel, Z. S., Nadgir, N., and Narayanan, A. AI agents that matter. *Transactions on Machine Learning Research*, 2024. arXiv:2407.01502.

Kuhn, L., Gal, Y., and Farquhar, S. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

Kumar, B., Lu, C., Gupta, G., Palepu, A., Bellamy, D., Raskar, R., and Beam, A. Conformal prediction with large language models for multi-choice question answering. *arXiv preprint arXiv:2305.18404*, 2023.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *NeurIPS*, 2017.

Oh, C. et al. Uncertainty quantification in LLM agents: Foundations, emerging challenges, and opportunities. *arXiv preprint arXiv:2602.05073*, 2026.

OpenAI. Introducing SWE-bench verified. OpenAI Blog, 2024. https://openai.com/index/introducing-swe-bench-verified/.

Quach, V., Fisch, A., Schuster, T., Yala, A., Sohn, J. H., Jaakkola, T. S., and Barzilay, R. Conformal language modeling. *ICLR*, 2024.

Rabanser, S., Kapoor, S., Kirgis, P., Liu, K., Utpala, S., and Narayanan, A. Towards a science of AI agent reliability. *arXiv preprint arXiv:2602.16666*, 2026.

Ramdas, A., Grünwald, P., Vovk, V., and Shafer, G. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 2023.

Ren, A. Z., Dixit, A., Bodrova, A., Singh, S., Chu, S., Snavely, N., Sadigh, D., and Angelova, A. Robots that ask for help: Uncertainty alignment for large language model planners. *CoRL*, 2023.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

Tian, K., Mitchell, E., Yao, H., Manning, C. D., and Finn, C. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *EMNLP*, 2023.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2023.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., and Hooi, B. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. *ICLR*, 2024.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Yao, S., Narasimhan, K., and Cao, Y. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Zhang, J., Choubey, P. K., Huang, K.-H., Xiong, C., and Wu, C.-S. Agentic uncertainty quantification. *arXiv preprint arXiv:2601.15703*, 2026.

Zhou, H., Van Landeghem, J., Popordanoska, T., and Blaschko, M. B. A novel characterization of the population area under the risk coverage curve (AURC) and rates of finite sample estimators. *ICML*, 2025.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Bisk, Y., Fried, D., Alon, U., et al. WebArena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

A. Gemini 3 Pro: Sampling Configuration

Gemini 3 Pro on our cloud API endpoint initially exhibited near-deterministic outputs at $T = 0.7$, producing identical action sequences in >99% of runs. We determined that the endpoint does not fully respect the `temperature` parameter without explicitly setting `top_p` < 1.0. After adding `top_p = 0.95`, outputs became properly stochastic (3.2 unique sequences on average, comparable to GPT-5). All Gemini results, including temperature ablation, use the corrected stochastic configuration.

B. Temperature Ablation: Full Results

Table 8. Full temperature ablation (20 questions, 5 runs each). $T = 0.7$ uses same questions from the main experiment.

Model	Temp	Accuracy	Unique Seqs
Llama 3.1 70B	0.0	82.0%	2.3
	0.3	83.0%	2.1
	0.7	82.0%	3.3
Claude Sonnet 4.5	0.0	85.0%	1.0
	0.3	87.0%	1.6
	0.7	85.0%	1.7
GPT-5	0.0	85.0%	1.4
	0.3	87.0%	1.5
	0.7	87.0%	1.6
Gemini 3 Pro	0.0	75.0%	1.0
	0.3	76.0%	2.0
	0.7	75.0%	2.2

C. Difficulty Stratification: Full Results

Task difficulty is computed as mean correctness across all runs from all four models. Strata: Easy (≥0.80), Medium (0.40–0.79), Hard (<0.40). Of 200 questions: 132 Easy, 32 Medium, 36 Hard.

Table 9. Consistency–correctness gap (pp) within difficulty strata. n_c/n_i = number of consistent/inconsistent tasks per model in that stratum.

Stratum	Llama	GPT-5	Claude	Gemini
Easy (132)	+11.2pp 46/58	+6.8pp 106/8	−0.5pp 119/9	+16.5pp 100/23
Medium (32)	+6.7pp 6/22	+15.0pp 16/10	+6.4pp 16/11	+34.3pp 9/21
Hard (36)	−5.6pp 9/24	+8.9pp 19/11	−5.8pp 20/12	−4.2pp 13/22

Partial correlations between unique sequences and accuracy, controlling for difficulty (continuous): Gemini $r = -0.487$ ($p < 0.001$), GPT-5 $r = -0.159$ ($p = 0.025$), Llama $r = -0.156$ ($p = 0.028$), Claude $r = 0.011$ ($p = 0.877$). The association remains significant for three of four models.

Consistent-Wrong analysis. The Hard stratum reversal (negative gaps above) is explained by *consistent-wrong* behavior: on tasks where the model lacks knowledge, consistency reflects confident commitment to an incorrect answer rather than reliable competence. We classify each model×Hard-task pair into four categories based on unique action sequences and correctness:

- **Consistent-Correct** (≤2 seqs, majority correct): 9/144 (6.3%)
- **Consistent-Wrong** (≤2 seqs, majority wrong): 52/144 (36.1%)
- **Inconsistent-Lucky** (≥4 seqs, ≥1 correct run): 27/144 (18.8%)
- **Inconsistent-Wrong** (≥4 seqs, 0 correct): 42/144 (29.2%)

Among consistent Hard tasks, 85.2% (52/61) are Consistent-Wrong: the model locks into one incorrect reasoning path. Of these, approximately 30% are false negatives under token F1 ≥ 0.5: the model produced a semantically correct answer in a different surface form (e.g., "Pasek and Paul" for gold "Pasek & Paul", "Kelly Osbourne" for gold "Kelly Lee Osbourne"). The remaining ∼70% are genuinely incorrect. When recomputed under token F1 ≥ 0.5, the Hard reversal disappears: all four models show non-negative consistency gaps (+9 to +16pp; Table 10), confirming that the consistency–correctness association holds under a more permissive metric. Conversely, 39.1% (27/69) of inconsistent Hard tasks achieve at least one correct run, confirming that behavioral variability enables exploration that occasionally recovers the right answer. Per-model consistent-wrong rates (fuzzy): Llama 100% (9/9), GPT-5 79% (15/19), Claude 80% (16/20), Gemini 92% (12/13).

Table 10. Hard stratum consistency gap (pp) under token F1 ≥ 0.5. All gaps are non-negative, confirming the fuzzy-match reversal is a metric artifact.

	Llama	GPT-5	Claude	Gemini
Gap (pp)	+8.9	+16.1	+9.7	+11.5
n_c/n_i	9/24	19/11	20/12	13/22

Table 11. HotpotQA vs. SWE-bench. Consistency ranking preserved across benchmark complexities, action spaces, and repositories.

	HotpotQA	SWE-bench
Action space	3 tools	Unrestricted bash
Tasks	200	50
Repositories	1 (Wikipedia)	5 (Python OSS)
Avg trajectory	3–8 steps	10–82 steps
Best model CV	8.2% (Claude)	17.7% (Claude)
Worst model CV	22.1% (Llama)	68.9% (Llama)
Models tested	4	4
Ranking preserved	✓ (Claude < GPT-5 < Gemini < Llama)	

D. Cross-Benchmark Comparison

E. GPT-4o: Additional Validation on 100-Question Subset

We additionally evaluated GPT-4o (OpenAI) on a 100-question subset (questions 1–100) collected when the model was available via our API provider. Results are consistent with the main findings: GPT-4o shows 2.5 unique sequences per task (comparable to GPT-5's 2.4), 73.3% accuracy, and a consistency–correctness gap of 35.9pp ($r = -0.47$, $p < 0.001$). Failure detection achieves AUROC 0.661, and the path length correlation is $\rho = 0.43$ ($p < 0.001$). Temperature ablation shows GPT-4o gains +9pp accuracy at $T = 0.0$, the largest temperature effect observed across all models tested. These results confirm that the consistency hierarchy extends to earlier-generation frontier models and that our findings are not specific to the four primary models in the main text.

F. Question Type Analysis

On the first 100 questions with type annotations, we compare bridge questions (multi-hop, $n = 79$) with comparison questions (yes/no, $n = 21$) using Llama 3.1 70B. Comparison questions show *higher* correctness (80.0% vs. 75.7%) but *lower* answer consistency (62.4% vs. 76.6%) and lower step variance (41% vs. 63%), highlighting that answer consistency and explanation consistency are distinct dimensions.

G. Metric Sensitivity

We verify robustness to the choice of correctness metric. Under exact match (EM), overall accuracy decreases (e.g., Claude: 43.6% EM vs. 81.5% fuzzy) but model rankings and consistency gaps are preserved. Under token F1 $>$ 0.5, results closely track fuzzy match. The consistency–correctness association holds across all three metrics for all four models.

H. Majority Voting and Selective Prediction: Full Results

Task-outcome taxonomy. Table 12 classifies each task by consistency ($\leq$2 vs. $\geq$4 unique sequences) and correctness (majority answer matches gold). The taxonomy explains why filtering outperforms voting: consistent-wrong tasks (5.5–10%) set the ceiling on filtering, while inconsistent-correct tasks (8–38%), where voting could theoretically consolidate the right answer, are rare except for Llama.

Table 12. Task-outcome taxonomy (% of tasks). Consistent: $\leq$2 unique seqs; Inconsistent: $\geq$4. Rows sum to $<$100% because tasks with exactly 3 unique sequences (6–17% per model) fall between thresholds.

Model	Cons-Right	Cons-Wrong	Inc-Right	Inc-Wrong
Claude Son. 4.5	67.5	10.0	10.5	5.5
GPT-5	61.5	9.0	7.5	7.0
Gemini 3 Pro	52.0	9.0	14.0	19.0
Llama 70B	25.0	5.5	38.0	14.0

Table 13 reports majority-vote accuracy across all budget levels. Gains are largest for Llama (+4.4pp at $k = 10$) and Claude (+2.0pp), while Gemini shows no improvement, consistent with the hypothesis that voting helps only when the model explores diverse-but-sometimes-correct paths.

Table 13. Majority-vote accuracy (%) by budget k (500 bootstrap iterations). Gain is relative to $k = 1$.

Model	$k=1$	$k=3$	$k=5$	$k=7$	$k=10$
Llama 70B	73.6	75.6+2.0	76.9+3.3	77.2+3.6	78.0+4.4
Claude S. 4.5	81.5	82.2+0.7	82.6+1.0	82.9+1.4	83.5+2.0
GPT-5	79.6	80.1+0.5	80.2+0.6	80.3+0.7	80.5+0.9
Gemini 3 Pro	72.2	72.3+0.1	72.2−0.0	72.2−0.0	71.7−0.4

Table 14 reports selective prediction at multiple agreement thresholds. Stricter thresholds yield higher accuracy but lower coverage. The accuracy–coverage tradeoff is favorable for Llama, Claude, and GPT-5; Gemini shows minimal improvement under selective prediction (+1.2pp at 3/3 unanimity), consistent with its high consistent-wrong rate relative to inconsistent-lucky rate.

Table 15 shows how the $k = 3$ majority-vote gain varies by difficulty stratum. Gains concentrate on Easy and Medium tasks; Hard tasks show no improvement, consistent with the

Table 14. Selective prediction: accuracy (%) and coverage (%) at varying agreement thresholds.

Model	Threshold	Acc.	Cov.
Llama 70B	Unanimous 3/3	87.2	53.8
	Majority 2/3	82.2	82.5
	Unanimous 5/5	88.7	44.2
	4/5	86.7	63.8
	Majority 3/5	83.6	80.3
Claude Son. 4.5	Unanimous 3/3	87.8	61.8
	Majority 2/3	84.4	84.4
	Unanimous 5/5	88.4	55.4
	4/5	87.8	67.3
	Majority 3/5	85.2	82.5
GPT-5	Unanimous 3/3	88.1	55.2
	Majority 2/3	83.2	78.8
	Unanimous 5/5	90.2	48.1
	4/5	86.9	62.3
	Majority 3/5	84.1	75.3
Gemini 3 Pro	Unanimous 3/3	73.4	64.7
	Majority 2/3	71.6	85.4
	Unanimous 5/5	74.3	58.4
	4/5	72.4	71.6
	Majority 3/5	71.6	83.2

finding that consistency on hard tasks reflects consistent-wrong behavior (Appendix C).

Why self-consistency works less in agentic settings. Wang et al. (2023) report 5–17pp gains from majority voting on single-turn CoT tasks (arithmetic, commonsense). Our agentic setting shows only 0–2pp gains. The difference is structural: in single-turn CoT, errors are primarily in the final reasoning step and the answer space is small, so diverse chains often converge on the correct answer. In multi-step agentic settings, an early wrong action (e.g., searching for the wrong entity at step 2) commits the agent to an incorrect trajectory for all subsequent steps, producing systematic errors that voting cannot correct. This makes consistency more valuable as a *diagnostic* signal (is the agent confident?) than as an *ensembling* strategy (pick the best answer).

Table 15. Per-stratum $k=3$ majority-vote gain (pp) over $k=1$.

Stratum	Llama	Claude	GPT-5	Gemini
Easy	+2.4	+0.3	+0.7	+0.7
Medium	+2.4	+1.8	+1.8	−1.2
Hard	−0.0	+0.9	−1.4	−0.5

I. Temperature Ablation

Figure 9 shows temperature ablation results across all four models. All models show monotonically increasing unique sequences with temperature. Claude and Gemini achieve near-perfect consistency at $T=0.0$ (1.0 sequences) while Llama retains 2.3, suggesting that model architecture matters beyond sampling noise. The finding that $T=0.0$ does not eliminate divergence for Llama or GPT-5 indicates that agentic inconsistency arises from factors beyond token-level sampling.

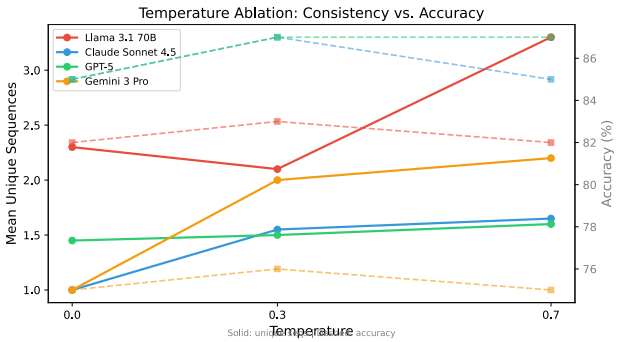


Figure 9. Temperature ablation (20 questions, 5 runs). Solid: unique sequences (left axis); dashed: accuracy (right axis).