FORMATTING INSTRUCTIONS FOR ICLR 2026 CONFERENCE SUBMISSIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) reasoning is typically evaluated using single runs, masking how much performance can vary across repeated executions. This practice obscures both reliability and cost, and can lead to misleading comparisons between reasoning methods and models. We introduce REASONBENCH, a benchmark suite and open-source library for controlled multi-run evaluation of LLM reasoning. For each model–strategy–task configuration, we perform repeated trials across 6 diverse benchmarks and report variance-aware metrics for both quality and cost, including confidence intervals and run-to-run variability measures. Using standardized implementations, we benchmark 10 widely used reasoning strategies under identical model conditions and evaluate 10 contemporary reasoning-oriented LLMs in a zero-shot setting. Our results show that run-to-run variability is substantial, benchmark-dependent, and often large enough to change model/method rankings relative to single-run averages. Additional analyses reveal that scaling within a model family improves both average quality and stability, while increasing test-time reasoning effort primarily increases cost without yielding statistically significant quality gains. Together, these findings motivate distribution-aware evaluation practices and provide reproducible tooling to support more reliable progress in LLM reasoning research. REASONBENCH is publicly available at `https://anonymous.4open.science/r/ReasonBench-64B3`.

1 INTRODUCTION

Recent studies highlight a growing tension between the promise of large language models (LLMs) and the risks of their adoption. Users tend to over-rely on AI-generated advice (Klingbeil et al., 2024), yet larger and more instructable models are becoming less reliable (Zhou et al., 2024b). Such instability may appear benign when the user wants more insights into `Muhammad Ali's` match history (cf. Fig. 1), but is dangerous in *safety-critical domains* such as medical decision-making, legal and financial reasoning, and autonomous systems, where unreliable outputs can carry severe consequences.

At the center of these concerns lies reasoning, which has become a primary frontier in the development of LLMs. Recent advances increasingly revolve around reasoning, whether through specialized strategies (Wei et al., 2022; Yao et al., 2023a; Klein et al., 2025), reasoning-focused training regimes such as DeepSeek R1 and OpenAI o1 (Guo et al., 2025; Jaech et al., 2024), or tool-augmented reasoning systems like Anthropic's Model Context Protocol and OpenAI's Deep Research. The demand for reliable reasoning is driven by impactful applications such as information seeking (Jin et al., 2025; Li et al., 2025), mathematical and formal logic reasoning (Yang et al., 2023; 2024a), and many other domains where structured problem solving is essential, making its robustness central to the field.

Traditionally, the behavior of machine learning algorithms has been framed through the bias–variance paradigm (Geman et al., 1992; Hastie et al., 2009), where bias captures systematic error and variance reflects run-to-run instability. Although this perspective has long guided classical ML, evaluations of LLMs, especially in reasoning tasks, focus almost exclusively on bias by reporting average accuracy from single or very few runs. Consequently, we lack statistically reliable performance estimates with confidence intervals, and instead rely on crude measurements that obscure the true instability of LLM reasoning, which, as shown in Fig. 1 (bottom), is substantial. For safety-critical applications in

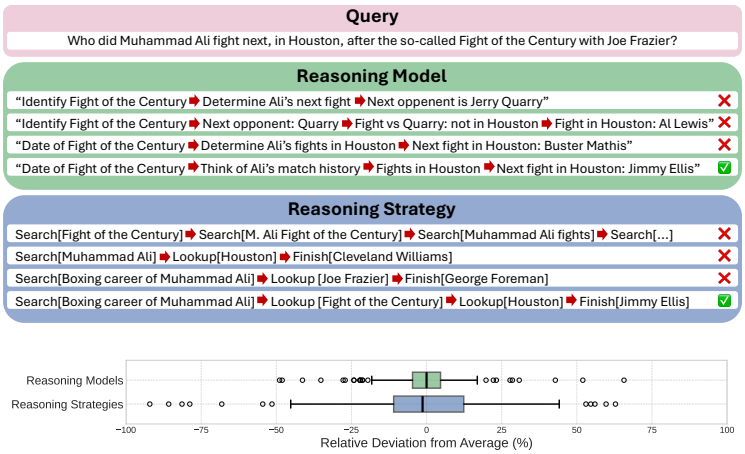


Figure 1: **Instability in LLM Reasoning.** For the same query, different reasoning models **(top)** and reasoning strategies **(middle)** produce distinct chains of thought and frequently contradictory conclusions. Even when working from identical instructions, methods vary widely in their intermediate reasoning steps and the correctness of their final answers. The **(bottom)** panel summarizes this variability quantitatively, showing that the relative deviation from average performance across reasoning models and strategies is massive.

particular, it is not only the mean accuracy that matters but also the lower bound of the confidence interval, which determines whether a system can be trusted in deployment.

Present Work. In this paper, we revisit the oldest trick in experimental science: repeat the experiment. We run 10 independent trials for each model–algorithm–task combination, reporting not only the mean but also the variance and confidence intervals of key performance metrics. Beyond evaluation, we release an agentic AI library (Fig. 2) that implements ten state-of-the-art reasoning algorithms and integrates with CacheSaver (Potamitis et al., 2025) for reproducible and cost-efficient experimentation, allowing us to establish reproducible baselines and provide practitioners with statistically reliable performance estimates.

Contributions.

• We introduce the **ReasonBench AI Library**, the first benchmark of 10 different LLM reasoning strategies across 4 different models and 6 different tasks with statistically reliable performance numbers (§ 2). Our framework offers a minimal, yet principled, abstraction layer over common patterns in agentic AI. By building on top of our API, researchers can implement new reasoning methods or tasks, through a guided evaluation framework, with only a few lines of code.

• We perform the **first systematic** multi-run evaluation of LLM reasoning strategies across diverse models and tasks (§ 3). Each model–algorithm–task combination is evaluated with ten independent runs, and we report statistically reliable estimates of quality and cost with confidence intervals.

• We conduct an **insight analysis** of the drivers of reasoning (in)stability (§ 4). Specifically, we study (i) **model scale** and its effect on both mean quality and run-to-run variability, (ii) the **impact of prompting**, (iii) **cost–quality correlations** across reasoning strategies, (iv) the effect of explicit **thinking-effort controls** on performance and stability, and (v) a **causal intervention on evaluation functions** showing improved performance and tighter confidence intervals for search-heavy methods.

• Based on our findings, we release a **leaderboard** that evaluates models through the **lens of stability** and propose **best practices and a call to action** for variance-aware evaluation in LLM reasoning research. The leaderboard is publicly available at `http://reasonbench.github.io`.

2 REASONBENCH

In this section, we provide a detailed description of our benchmarking framework, REASONBENCH, which we release as both a benchmark suite and an open-source AI library. REASONBENCH is

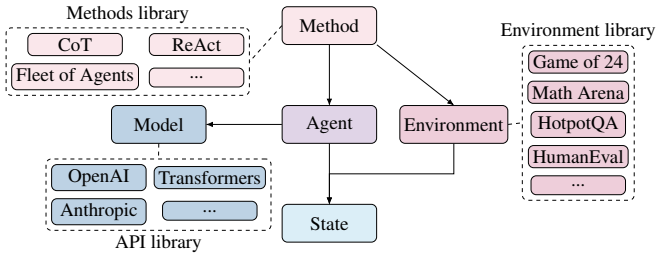


Figure 2: **ReasonBench architecture.** Methods orchestrate the three core components: Agents, Environments, and Models. Agents translate states into prompts, query models, and parse responses into actions. Environments are drawn from a large task library and offer functions such as next-step transitions and scoring heuristics. Models provide a unified interface to external LLM APIs. States record the intermediate configurations of reasoning, enabling reproducibility and fair comparison across tasks and methods.

designed with three goals in mind: (i) principled implementations of diverse reasoning strategies, (ii) reproducible and cost-efficient experimentation, and (iii) extensibility so the community can easily contribute new methods, models, or tasks.

2.1 LIBRARY DESIGN

REASONBENCH is organized around a set of core abstractions that capture the building blocks of reasoning pipelines. The principal components are the Method, Environment, Agent, State, and Model, which together define a modular interface for implementing reasoning algorithms, connecting to LLMs, and interacting with tasks. In designing these components, we followed established principles from software architecture engineering, emphasizing modularity, separation of concerns, and extensibility. Figure 2 illustrates the relationships between these abstractions.

Method Abstraction. The method abstraction specifies the overall logic of a reasoning strategy independently of the underlying model or task. A method integrates agents, which construct prompts and parse responses; the environment, which maintains and updates the task state; and the model, which produces candidate outputs. Each method exposes a standard interface for solving tasks by generating and updating sequences of states, and a benchmarking routine that runs multiple problem instances in parallel. This makes methods interchangeable and extensible: once the interface is implemented, a new reasoning algorithm can be evaluated consistently across models, tasks, and metrics within the benchmarking pipeline.

Environment Abstraction. The environment abstraction formalizes the task-specific dynamics of reasoning. It governs how a state evolves in response to an action, how to determine whether an action is valid, when a trajectory has reached a terminal condition, and how to evaluate the final outcome. By encapsulating these rules, the environment decouples domain logic from reasoning algorithms, allowing the same method to be applied consistently across tasks while ensuring that actions and evaluations remain faithful to each benchmark.

Agent Abstraction. The agent abstraction defines the interface between methods, models, and states. Agents specify how prompts are constructed from the current state, how queries are issued to the model, and how responses are parsed into actions that update the environment. This unified interface makes it possible to express a wide spectrum of reasoning strategies: from simple input–output prompting to multi-step reasoning, search procedures, candidate aggregation, and self-evaluation. By isolating prompt construction and response handling, ReasonBench supports diverse reasoning paradigms without altering the abstractions for methods, environments, or models.

State Abstraction. The state abstraction captures the intermediate configuration of a reasoning process. It provides a standardized way to represent progress on a task and to handle states with controlled randomness. Methods interact only with states, while environments define how actions modify them and how final outcomes are assessed. This separation ensures that reasoning trajectories can be reproduced, compared, and analyzed independently of the underlying task domain.

Table 1: **Quality and cost variability of contemporary reasoning models across all benchmarks.** Gemini-3 Flash achieves the strongest and most stable quality overall, though at the highest cost, while GPT-5 Mini offers competitive performance with minimal cost. GPT-4.1 Nano and DeepSeek-R1 show the worst performance with the latter having the worst overall variability in terms of both quality and cost. The best performance is shown in **blue** whereas the worst is shown in **orange**.

Reasoning Model	Provider	Quality				Cost			
		Average*	Run Deviation*	Noise (Global)	Noise (Run)	Average*	Run Deviation*	Noise (Global)	Noise (Run)
DeepSeek R1	DeepSeek	0.2217 [0.20, 0.25]	17.81% [0.083, 0.324]	**1.1096**	0.0381	**1.3141 [1.27, 1.36]**	4.69% [0.019, 0.078]	**0.7978**	**0.0168**
Llama 4 Maverick	Meta	0.4029 [0.38, 0.43]	8.27% [0.035, 0.162]	0.3797	0.0358	0.0186 [0.02, 0.02]	3.24% [0.013, 0.057]	0.0025	0.0000
GPT-4.1 mini	OpenAI	0.4540 [0.43, 0.48]	10.74% [0.070, 0.151]	0.8653	0.0205	0.0145 [0.01, 0.02]	**8.9% [0.034, 0.188]**	**0.0023**	0.0001
GPT-4.1 nano	OpenAI	**0.1063 [0.10, 0.12]**	13.66% [0.055, 0.229]	0.1957	**0.0153**	**0.0054 [0.01, 0.01]**	**2.56% [0.008, 0.052]**	0.0116	**0.0000**
Qwen3 235B Thinking	Alibaba	0.4124 [0.39, 0.43]	**39.38% [0.193, 1.599]**	0.6612	0.0301	0.5366 [0.52, 0.56]	4.9% [0.022, 0.082]	0.1082	0.0038
GPT-OSS 120B	OpenAI	0.5025 [0.47, 0.53]	9.84% [0.035, 0.174]	**0.1331**	0.0479	0.0304 [0.03, 0.03]	5.69% [0.025, 0.097]	0.0038	0.0000
GPT-5 mini	OpenAI	0.5644 [0.53, 0.60]	9.5% [0.046, 0.156]	0.2456	0.0531	0.1674 [0.16, 0.18]	4.76% [0.021, 0.078]	0.0593	0.0004
GPT-5 nano	OpenAI	0.5048 [0.48, 0.52]	10.78% [0.061, 0.169]	0.6089	0.0348	0.0591 [0.06, 0.06]	3.84% [0.017, 0.063]	0.0078	0.0000
Claude Haiku 4.5	Anthropic	0.3777 [0.36, 0.40]	11.7% [0.033, 0.228]	0.2485	**0.0537**	0.1099 [0.10, 0.12]	3.7% [0.013, 0.074]	0.0052	0.0007
Gemini 3 Flash	Google	**0.7810 [0.74, 0.78]** †	**3.48% [0.015, 0.054]**	0.2363	0.0345	1.0451 [0.98, 1.05]	3.38% [0.015, 0.054]	0.3411	0.0124

† Indicates statistical significance ($p < 0.05$) between the best and the second-best scores.
* Reports average value and 95% confidence intervals in brackets.
Note: Models are ordered by release date (2025). Dashed horizontal rules indicate models released in the same quarter.

Model Abstraction. The model abstraction provides a uniform interface for interacting with language models, supporting both single and batched queries across diverse providers. Built on top of asynchronous execution (via *asyncio*) and integrated with response caching through *CacheSaver*, it is both extensible and accountable: new models can be added without modifying the framework, and every interaction logs latency, token usage, and generation metadata. This combination enables deterministic reproducibility across repeated experiments while distinguishing between newly generated, reused, and deduplicated outputs.

2.2 Experimental Setup

Number of runs. We repeat all experiments 10 times and report both mean and confidence intervals of the evaluation metrics.

Tasks and data. We evaluate on six benchmark tasks selected to cover a broad spectrum of reasoning, planning, and general problem-solving abilities. These tasks span diverse domains: (1) mathematical reasoning: Game of 24 (Yao et al., 2023a) (2) coding: HumanEval (Chen et al., 2021), (3) question answering and knowledge reasoning: HotpotQA (Zhilin et al., 2018) and Humanity's Last Exam (Phan et al., 2025), (4) scientific reasoning: SciBench (Wang et al., 2024a), and (5) creative writing: Shakespearean Sonnet Writing (Suzgun & Kalai, 2024). For consistency, we rely on the test sets released with the original benchmarks.

Reasoning strategies. We experiment with 10 representative state-of-the-art reasoning strategies: (1) IO prompting, (2) CoT, (3) CoT-SC, (4) React (Yao et al., 2023b), (5) Reflexion, (6) ToT-DFS (Yao et al., 2023a), (7) TOT-BFS (Yao et al., 2023a), (8) GoT, (9) RAP (Hao et al., 2023), and (10) FoA (Klein et al., 2025). To ensure that comparisons between methods are *fair*, each strategy has been re-implemented within ReasonBench using a standardized interface, which harmonizes prompt handling, state transitions, and evaluation. Our selection criterion requires that methods provide publicly available code for at least one of the tasks considered in this study. Consequently, we exclude TouT (Mo & Xin, 2024), and RecMind (Wang et al., 2024b). We also omit BoT (Yang et al., 2024b), where the code is released but a key resource (the meta-buffer) is missing, preventing reproducibility. LATS (Zhou et al., 2024a) is excluded due to its prohibitive computational cost.

Reasoning models. We evaluate a diverse set of contemporary reasoning models spanning multiple providers: (1) OpenAI: GPT-OSS-120B Agarwal et al. (2025), GPT-4.1 Mini, GPT-4.1 Nano, GPT-5 Mini, GPT-5 Nano, (2) DeepSeek: DeepSeek R1 Guo et al. (2025), (3) Meta: Llama 4 Scout AI (2025), (4) Alibaba Cloud Qwen3-235B Yang et al. (2025a), (5) Google: Gemini-3 Flash Comanici et al. (2025) and (6) Claude-Haiku 4.5. These models represent the latest generation of systems that aim to perform end-to-end reasoning, without requiring explicit scaffolding through external frameworks. To ensure comparability, all models are evaluated in a zero-shot setting using identical

Table 2: **Quality and cost variability across reasoning frameworks under GPT-4.1 Nano.** Direct methods show low cost but high instability in quality, while structured and planning-based approaches incur higher cost with mixed consistency. FoA delivers the most stable performance overall, whereas RAP and ToT-BFS exhibit the highest noise across benchmarks and runs respectively. The best performance is shown in **blue** whereas the second best is shown in **orange**.

Strategy	Type	Quality				Cost			
		Average*	Run Deviation*	Noise (Global)	Noise (Run)	Average*	Run Deviation*	Noise (Global)	Noise (Run)
IO	Direct	**0.1063 [0.10, 0.12]**	13.66% [0.055, 0.229]	**0.1957**	**0.0153**	**0.0054 [0.01, 0.01]**	2.56% [0.008, 0.052]	0.0116	**0.0000**
CoT Wei et al. (2022)	Direct	0.2761 [0.25, 0.30]	29.59% [0.152, 0.492]	0.6016	0.0940	0.0130 [0.01, 0.01]	4.73% [0.026, 0.072]	**0.0111**	0.0000
CoT-SC Wang et al. (2023)	Direct	0.2281 [0.21, 0.24]	**65.54% [0.349, 1.809]**	0.3187	0.0427	0.0682 [0.07, 0.07]	**0.74% [0.003, 0.012]**	0.0649	0.0000
ReAct Yao et al. (2023b)	Adaptive	0.2956 [0.28, 0.31]	29.14% [0.177, 0.704]	1.1289	0.0219	0.0697 [0.07, 0.07]	**6.45% [0.027, 0.125]**	0.0235	0.0002
Reflexion Shinn et al. (2023)	Adaptive	0.2815 [0.27, 0.30]	27.75% [0.146, 0.458]	1.3080	0.0413	0.1647 [0.15, 0.17]	4.79% [0.037, 0.061]	0.0315	0.0007
ToT-DFS Yao et al. (2023a)	Structured	0.1272 [0.10, 0.14]	5.15% [0.012, 0.112]	1.2396	0.0353	0.1033 [0.10, 0.11]	3.55% [0.013, 0.059]	0.3541	0.0059
ToT-BFS Yao et al. (2023a)	Structured	0.4073 [0.38, 0.44]	14.35% [0.054, 0.232]	0.4816	**0.1781**	0.4428 [0.43, 0.46]	4.82% [0.023, 0.081]	0.9883	**0.0064**
GoT Besta et al. (2024)	Structured	0.3361 [0.31, 0.36]	15.64% [0.068, 0.279]	0.5101	0.1203	0.4971 [0.48, 0.51]	1.81% [0.009, 0.029]	1.2939	0.0025
RAP Hao et al. (2023)	Planning	0.3669 [0.35, 0.38]	18.54% [0.117, 0.417]	**1.5461**	0.0273	**0.5320 [0.52, 0.54]**	4.19% [0.008, 0.096]	**1.6642**	0.0021
FoA Klein et al. (2025)	Evolutionary	**0.4580 [0.43, 0.48]** †	**7.83% [0.030, 0.173]**	0.4716	0.1522	0.3237 [0.32, 0.33]	3.75% [0.016, 0.061]	0.3221	0.0010

† Indicates statistical significance ($p < 0.05$) between the best and the second-best scores.
* Reports average value and 95% confidence intervals in brackets.

benchmark prompts, with decoding parameters harmonized across providers. Our selection criterion prioritizes flagship reasoning-oriented releases from major labs that are accessible via public APIs at the time of writing.

Evaluation metrics. We evaluate methods along two dimensions: *quality* and *cost*. For each dimension, we report four complementary metrics. *Average* performance is estimated using a stratified bootstrap over runs, where each benchmark is treated as a stratum and confidence intervals reflect expected performance under reruns of the same benchmark suite. *Run Deviation* measures typical run-to-run deviation from a strategy's mean on each benchmark, computed as a bootstrapped average of normalized absolute errors. To quantify stochasticity independent of benchmark difficulty, we additionally report two noise metrics based on z-score normalization: *Noise (Global)*, defined as the variance of all z-scores across benchmarks, and *Noise (Run)*, defined as the average within-benchmark z-score variance. Cost metrics (token usage and wall-clock runtime) are reported using the same statistics and expressed in USD based on the provider's pricing.

3 Experiments

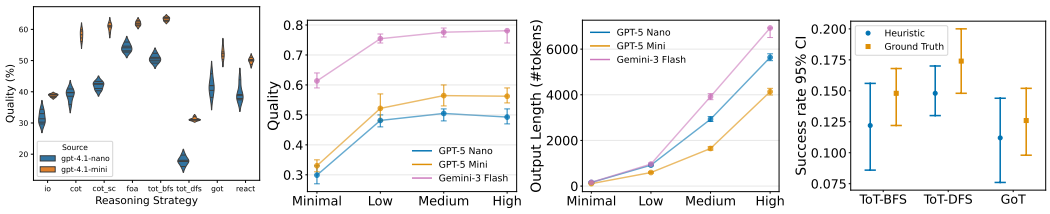
Our results are structured around two complementary questions: (i) how do different reasoning strategies compare when applied under identical model conditions, and (ii) how do different reasoning models perform when asked to solve benchmarks directly without additional framework support. To answer the first question, we fix GPT-4.1-Nano as the underlying model and evaluate ten representative reasoning strategies across all benchmarks. To address the second, we evaluate ten open- and closed-source reasoning models, from 6 diverse model providers, in a zero-shot setting, measuring their ability to solve tasks without external scaffolding. The resources for reproducing our experiments are available at `https://anonymous.4open.science/r/ReasonBench-64B3`.

3.1 Reasoning strategies

Table 2 summarizes suite-level quality and cost metrics, along with their stability measures, for ten reasoning *strategies* evaluated under identical model conditions. Five results stand out.

Quality generally increases with cost. For strategies, quality generally increases with cost. Unlike reasoning models (Table 1), where cost and quality are decoupled, strategy cost reflects deliberate compute investment (branching, sampling, search) that often translates to accuracy gains.

Top-performing strategies are cost-efficient. FoA achieves the highest average quality, followed by ToT-BFS. Both sit far above direct prompting baselines in cost, but are also not the most expensive ones, indicating that the quality frontier here is dominated by strategies that do substantially more work per query.



(a) (b) (c) (d)

Figure 3: **(a) Model scale and stability:** GPT-4.1 MINI yields higher quality and tighter distributions than GPT-4.1 NANO across strategies. **(b–c) Thinking effort:** increasing the reasoning effort sharply increases output tokens while quality saturates across GPT-5 NANO, GPT-5 MINI, and GEMINI-3 FLASH. **(d) Causal evaluation intervention:** on *Game of 24*, replacing the heuristic evaluator with ground-truth evaluation changes success rates and improves the stability in most cases.

Within-task stochasticity varies dramatically. The noise of quality across runs spans more than an order of magnitude. Thus, the *strategy* itself can be a primary driver of repeat-run variability, especially for high-performing methods such as ToT-BFS.

Adaptive and planning strategies show strong benchmark sensitivity. Quality Global Noise spans nearly an order of magnitude. These elevated values indicate uneven standardized performance across benchmarks. As a result, a strategy's overall average can shift substantially as the benchmark mix changes.

Quality and cost stability decouple. Cost variability does not reliably mirror quality variability: CoT-SC has the worst quality run deviation while exhibiting the lowest cost run deviation, while ReAct has the highest cost run deviation without being the most quality-unstable. Structured/planning methods also show large cost Global Noise, implying that bills can be highly task-dependent even when the mean cost is stable.

3.2 REASONING MODELS

Having established that strategies can strongly modulate both accuracy and instability, Table 1 holds the strategy fixed and compares ten contemporary *models* across providers using the same metrics. We highlight five results.

Cost does not correlate with quality. Cost and quality are weakly related in Table 1. The two most expensive models illustrate this starkly: DeepSeek R1 has the highest cost but ranks second to last in quality, while Gemini 3 Flash is similarly expensive but best in quality.

The highest-quality model is also the most repeatable. Similar to the reasoning strategies, the best performing model, Gemini 3 Flash achieves the highest quality average and the lowest quality run deviation, contradicting a simple quality–variance trade-off at the top end.

Instability is multi-dimensional. The noise quality between runs lies in a relatively narrow band, whereas the run deviation from the average spans an order of magnitude. Thus, large differences in apparent instability are driven more by score-scale and benchmark interaction than by radically different intrinsic stochasticity.

Global Noise captures benchmark-dependence. Global Noise varies widely, separating generalists from benchmark-sensitive models. GPT-OSS 120B has the lowest Global Noise (0.133), while DeepSeek R1 (1.110) and GPT-4.1 nano (1.049) are the highest, indicating a strong task sensitivity.

Quality stability and cost stability decouple. Quality and cost variability are only weakly related. For example, GPT-4.1 mini has a moderate quality run deviation (10.74%) but the worst cost run deviation (8.9%), while GPT-5 nano has a stable cost (3.84%) despite moderate quality variability (10.78%).

4 ANALYSIS

4.1 SCALING EFFECTS WITHIN A MODEL FAMILY

In Figure 3a, we analyze the stability of reasoning performance within a single model family at different scales. We consider GPT-4.1-Nano and GPT-4.1-Mini, evaluating them on all benchmarks with ten independent runs.

Model Scaling Improves Accuracy and Stability. Across all strategies, we observe a consistent scaling effect: GPT-4.1-Mini (gold violins) achieves higher mean quality and exhibits substantially tighter distributions than GPT-4.1-Nano (blue violins). This indicates that increasing model size within the same family not only improves average performance but also reduces Noise (Run), leading to more stable reasoning behavior overall.

4.2 IMPACT OF PROMPTS ON STABILITY

We investigate whether instability in reasoning performance arises from prompts and parsers, in addition to the reasoning strategies themselves. Prompting artifacts, such as underspecified answer formats, amplify stochastic variation and cause divergent outputs across runs. If such interface-level choices drive instability, then improving prompt clarity and parsing robustness should reduce variability without altering the underlying reasoning logic. The results are reported in Table 5.

Prompt Refinements Improve Quality but Not Stability. Across all strategies, clarifying prompts and strengthening the parsing logic leads to statistically significant improvements in average quality, while run-to-run variance remains largely unchanged. This indicates that existing reasoning methods are sensitive to prompt specification artifacts, and that prompt-level refinements can systematically improve outcomes without altering the underlying reasoning strategy.

4.3 CORRELATION BETWEEN QUALITY AND COST

Finally, we analyze the relationship between variability in quality and cost. Our approach allows us to examine multiple outcomes at the level of individual samples, recording quality and the exact cost incurred for each run and benchmark. Results are shown in Figure 8.

Divergent Cost–Quality Relationships. Reasoning strategies differ fundamentally in how additional computation translates into quality, with some benefiting from higher cost, others degrading, and some exhibiting task-dependent behavior. In particular, FoA shows a consistently positive association, with higher-cost samples yielding higher-quality output, indicating stable scaling behavior. In contrast, ReAct exhibits a negative slope across all tasks, where increased computation corresponds to less reliable reasoning. GoT shows no consistent trend, reflecting sensitivity to task structure.

4.4 THINKING EFFORT

In Figs. 3b and 3c we study the effect of explicit reasoning effort controls on model performance and stability. We evaluate three models that can control how many reasoning tokens to generate before creating a response to the prompt: GPT-5-nano, GPT-5-mini, and Gemini 3 Flash.

More for less. Increasing reasoning effort consistently increases computational cost, while accuracy gains are limited and non-monotonic. Performance improves from low to medium effort and then saturates or declines, despite sharply higher token usage, consistent with the inverted U-shaped relationship between the reasoning length and the accuracy Wu et al. (2025). However, across multiple independent runs, a new story unfolds as differences in average performance are not statistically significant.

4.5 INTERVENING ON EVALUATION FUNCTIONS

Many reasoning strategies rely on an evaluation function to estimate progress toward a solution. To test the causal role of this signal, we perform a controlled intervention on the *Game of 24* task,

which allows exact ground-truth evaluation of intermediate states. We benchmark the same reasoning strategy with the heuristic evaluation function but also by replacing it with the ground-truth evaluation.

Causal effect of evaluation functions. Replacing heuristic evaluation with ground-truth evaluation on *Game of 24* consistently improves reasoning performance (Fig. 3d) and stability across all strategies (Fig. 9). Ground-truth evaluation yields higher average quality, lower score variance, and tighter confidence intervals, with the largest gains observed for search-heavy methods such as ToT-BFS.

5 Discussion

5.1 Implications and Broader Impact

Prefer reasoning models over external strategies. Reasoning-capable models are often the preferable default: they exhibit substantially lower variance than external reasoning strategies and typically provide a better quality–cost trade-off, whereas strategy-level gains are frequently accompanied by higher instability and less predictable spend.

Reasoning effort scales cost, not quality. Increasing the reasoning effort during test-time, consistently and markedly raises cost, yet yields limited and mostly statistically indistinguishable improvements in average performance.

Global Noise reveals benchmark-dependence. Global Noise varies widely across systems, indicating that some systems are markedly more task-sensitive even when aggregate scores appear competitive; benchmark-dependence should therefore be treated as a first-class reliability issue.

5.2 Recommendations to the Community

Adaptive, variance-aware reasoning stacks. Search-heavy strategies can improve mean performance while increasing run-to-run variance, and no single method is uniformly best across tasks. A practical direction is to build *compositional* pipelines with a learned *router* that gates between simple decoding and structured search using problem features or early signals, trained to minimize variance under constraints.

Distribution-aware benchmarking (beyond best-of-k). Best-of-k reporting is brittle in high-variance regimes and can alter the reasoning system's rankings as k changes. Benchmarks should emphasize distributional reporting and avoid best-of-k as a primary leaderboard criterion.

Evaluators and Stability Diagnostics. Evaluator quality can be causal for both accuracy and stability in search, so verifiers should be calibrated and uncertainty-aware, with explicit study of how evaluator noise propagates into the downstream variance. More broadly, we advocate an analysis lens for reasoning that probes into branching decisions and verifier signals that could localize where stochastic divergence begins and guide stability-oriented design.

5.3 Limitations and Future Work

Limitations. We investigate decoding stochasticity, while ignoring other variability sources. Then, our benchmark suite, strategy set, and evaluated reasoning models are representative but incomplete. Broader domain coverage across all dimensions can strengthen our findings and potentially unveil even more elusive findings. Moreover, while we study prompt and parsing sensitivity, we do not evaluate more advanced prompt interventions, such as automatic prompt optimization, and their effects on variability.

Future work. Future work should move from measuring instability to *optimizing* it: develop stability-aware training and selection objectives that penalize variance rather than optimizing mean accuracy alone. Additionally, learn adaptive routers that jointly choose the reasoning method and compute budget to satisfy reliability constraints under cost limits. Finally, extending ReasonBENCH to tool-using agents would test stability under external nondeterminism.

IMPACT STATEMENT

This work argues that *stability is a first-class property of LLM reasoning*, not a peripheral evaluation detail. REASONBENCH operationalizes this view by making run-to-run variability measurable and comparable for both quality and cost, using controlled multi-run protocols and variance-aware metrics. The immediate impact is methodological: it enables more faithful scientific conclusions by revealing when apparent improvements in mean performance are not robust, when cost and quality decouple, and when benchmark dependence can dominate aggregate scores. Practically, these measurements support safer and more predictable deployment by highlighting lower-bound behavior, budget/latency uncertainty, and task sensitivity that are invisible to single-run reporting.

Beyond measurement, REASONBENCH provides reusable tooling that lowers the barrier to adopting variance-aware evaluation, helping the community reproduce results, rerun experiments as pipelines evolve, and compare methods under standardized implementations. We expect this to reduce over-claiming and "leaderboard churn" driven by noise, and to incentivize research on models and methods that are reliable under realistic constraints. A potential negative impact is increased evaluation cost from repeated trials; however, the ability to share standardized pipelines, cache intermediate artifacts, and reuse multi-run statistics can amortize this overhead and make reliability checks routine rather than exceptional.

Finally, the broader implications of our findings are not merely observational: the Discussion (§ 5) of the main paper includes dedicated *Implications and Broader Impact* (§ 5.1) and *Recommendations to the Community* (§ 5.2) sections that translate empirical results into actionable guidance. We hope REASONBENCH catalyzes a shift toward distribution-aware reporting standards and stability-oriented design, accelerating progress toward reasoning systems that are not only accurate on average but stable, reproducible, and dependable in practice.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*, 2024.

Meta AI. Introducing llama 4: Advancing multimodal intelligence, 2025. Accessed: 2025-09-22.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *AAAI*, volume 38, pp. 17682–17690, 2024.

Robert E Blackwell, Jon Barry, and Anthony G. Cohn. Towards reproducible llm evaluation: Quantifying uncertainty in llm benchmark scores. *ArXiv*, abs/2410.03492, 2024.

Chen et al. Evaluating large language models trained on code, 2021. arXiv eprint 2107.03374, cs.LG.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *EMNLP*, 2023.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, et al. The elements of statistical learning, 2009.

Fabian Hoppe, Filip Ilievski, and Jan-Christoph Kalo. Investigating the robustness of deductive reasoning with large language models. *arXiv preprint arXiv:2502.04352*, 2025.

Yutao Hou, Zeguan Xiao, Fei Yu, Yihan Jiang, Xuetao Wei, Hailiang Huang, Yun Chen, and Guanhua Chen. Automatic robustness stress testing of llms as mathematical problem solvers. *arXiv preprint arXiv:2506.05038*, 2025.

Shulin Huang, Linyi Yang, Yan Song, Shuang Chen, Leyang Cui, Ziyu Wan, Qingcheng Zeng, Ying Wen, Kun Shao, Weinan Zhang, et al. Thinkbench: Dynamic out-of-distribution evaluation for robust llm reasoning. *arXiv preprint arXiv:2502.16268*, 2025.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Kaiyuan Ji, Yijin Guo, Zicheng Zhang, Xiangyang Zhu, Yuan Tian, Ning Liu, and Guangtao Zhai. Medomni-45 {\deg}: A safety-performance benchmark for reasoning-oriented llms in medicine. *arXiv preprint arXiv:2508.16213*, 2025.

Enyi Jiang, Changming Xu, Nischay Singh, and Gagandeep Singh. Misaligning reasoning with answers–a framework for assessing llm cot robustness. *arXiv preprint arXiv:2505.17406*, 2025.

Bowen Jin, Jinsung Yoon, Priyanka Kargupta, Sercan O Arik, and Jiawei Han. An empirical study on reinforcement learning for reasoning-search interleaved llm agents. *arXiv preprint arXiv:2505.15117*, 2025.

Rushang Karia, Daniel Bramblett, Daksh Dobhal, and Siddharth Srivastava. Autonomous evaluation of llms for truth maintenance and reasoning tasks. *arXiv preprint arXiv:2410.08437*, 2024.

Lars Henning Klein, Nearchos Potamitis, Roland Aydin, Robert West, Caglar Gulcehre, and Akhil Arora. Fleet of agents: Coordinated problem solving with large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Artur Klingbeil, Cassandra Grützner, and Philipp Schreck. Trust and reliance on ai—an experimental study on the extent and costs of overreliance on ai. *Computers in Human Behavior*, 160:108352, 2024.

Yangning Li, Weizhi Zhang, Yuyao Yang, Wei-Chieh Huang, Yaozu Wu, Junyu Luo, Yuanchen Bei, Henry Peng Zou, Xiao Luo, Yusheng Zhao, et al. Towards agentic rag with deep reasoning: A survey of rag-reasoning systems in llms. *arXiv preprint arXiv:2507.09477*, 2025.

Junnan Liu, Hong wei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang, and Kai Chen. Are your llms capable of stable reasoning? In *Annual Meeting of the Association for Computational Linguistics*, 2024.

Lovish Madaan, Aaditya K Singh, Rylan Schaeffer, Andrew Poulton, Sanmi Koyejo, Pontus Stenetorp, Sharan Narang, and Dieuwke Hupkes. Quantifying variance in evaluation benchmarks. *arXiv preprint arXiv:2406.10229*, 2024.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024.

Moran Mizrahi, Guy Kaplan, Daniel Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2023.

Shentong Mo and Miao Xin. Tree of uncertain thoughts reasoning for large language models. In *ICASSP*, pp. 12742–12746, 2024.

Philipp Mondorf and Barbara Plank. Beyond accuracy: evaluating the reasoning behavior of large language models–a survey. *arXiv preprint arXiv:2404.01869*, 2024.

Jinjie Ni, Fuzhao Xue, Xiang Yue, Yuntian Deng, Mahir Shah, Kabir Jain, Graham Neubig, and Yang You. Mixeval: Deriving wisdom of the crowd from llm benchmark mixtures. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 98180–98212. Curran Associates, Inc., 2024.

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

Nearchos Potamitis, Lars Henning Klein, Bardia Mohammadi, Chongyang Xu, Attreyee Mukherjee, Niket Tandon, Laurent Bindschaedler, and Akhil Arora. Cache saver: A modular framework for efficient, affordable, and reproducible LLM inference. In *EMNLP*, pp. 25703–25724, 2025. doi: 10.18653/v1/2025.findings-emnlp.1402.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *NeurIPS*, pp. 8634–8652, 2023.

Mirac Suzgun and Adam Tauman Kalai. Meta-prompting: Enhancing language models with task-agnostic scaffolding. *arXiv preprint arXiv:2401.12954*, 2024.

Ganghua Wang, Zhaorun Chen, Bo Li, and Haifeng Xu. Cer-eval: Certifiable and cost-efficient evaluation framework for llms. *arXiv preprint arXiv:2505.03814*, 2025.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: evaluating college-level scientific problem-solving abilities of large language models. In *ICML*, 2024a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. Recmind: Large language model powered agent for recommendation. In *NAACL-HLT (Findings)*, pp. 4351–4364, 2024b.

Yuqing Wang and Yun Zhao. Rupbench: Benchmarking reasoning under perturbations for robustness evaluation in large language models. *arXiv preprint arXiv:2406.11020*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Kaiyu Yang, Aidan Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan J Prenger, and Animashree Anandkumar. Leandojo: Theorem proving with retrieval-augmented language models. *Advances in Neural Information Processing Systems*, 36:21573–21612, 2023.

Kaiyu Yang, Gabriel Poesia, Jingxuan He, Wenda Li, Kristin Lauter, Swarat Chaudhuri, and Dawn Song. Formal mathematical reasoning: A new frontier in ai. *arXiv preprint arXiv:2412.16075*, 2024a.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E. Gonzalez, and Bin Cui. Buffer of thoughts: Thought-augmented reasoning with large language models. In *NeurIPS*, 2024b.

Yuli Yang, Hiroaki Yamada, and Takenobu Tokunaga. Evaluating robustness of llms to numerical variations in mathematical reasoning. In *The Sixth Workshop on Insights from Negative Results in NLP*, pp. 171–180, 2025b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023a.

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

Fanghua Ye, Mingming Yang, Jianhui Pang, Longyue Wang, Derek Wong, Emine Yilmaz, Shuming Shi, and Zhaopeng Tu. Benchmarking llms via uncertainty quantification. *Advances in Neural Information Processing Systems*, 37:15356–15385, 2024.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. ReST-MCTS*: LLM self-training via process reward guided tree search. In *NeurIPS*, 2024.

Yang Zhilin, Qi Peng, Zhang Saizheng, Bengio Yoshua, Cohen William, Salakhutdinov Ruslan, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018. doi: 10.18653/v1/d18-1259.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *ICML*, 2024a.

Lexin Zhou, Wout Schellaert, Fernando Martínez-Plumed, Yael Moros-Daval, Cèsar Ferri, and José Hernández-Orallo. Larger and more instructable language models become less reliable. *Nature*, 634(8032):61–68, 2024b.

A Related works

Instability in LLM Reasoning. A growing body of work highlights that LLM reasoning can be brittle and unstable. Benchmarks such as (Jiang et al., 2025; Wang & Zhao, 2024) show that small lexical or semantic changes to inputs can cause inconsistent reasoning chains and consequently large drops in performance. Similar insights emerge from perturbation studies in deductive logic and mathematics, including (Hoppe et al., 2025) and (Yang et al., 2025b). Beyond perturbations, survey work such as (Ahn et al., 2024) documents that models often arrive at different answers for identical problems via divergent reasoning paths. Stress-test frameworks such as (Hou et al., 2025) and (Huang et al., 2025) generate adversarial or out-of-distribution prompting variants to reveal systematic weaknesses in mathematical and commonsense reasoning. Across studies, the findings point to an endemic problem: LLM reasoning is highly sensitive to perturbations and randomness, making reproducibility an open problem.

Calls for Better Evaluation Practices. Alongside these studies, researchers are emphasizing the need for more rigorous evaluation methodologies. Miller (2024) summarizes the best-practice methodology from a statisticians toolbox and provides LLM-focused guidelines on reporting uncertainty, advocating for confidence intervals, clustered standard errors, and statistical tests based on question-level paired differences. Similar calls appear in (Mizrahi et al., 2023), which demonstrates the sensitivity of results to prompt wording, and in (Ni et al., 2024), which argues for aggregating across benchmarks to reduce instability. (Blackwell et al., 2024) argues that, even on simple QA benchmarks, repeated runs are required to reach statistically reliable conclusions. Survey contributions such as (Mondorf & Plank, 2024) echo this perspective, arguing that focusing on shallow accuracy metrics obscures important behavioral properties. Collectively, these works call for reproducibility, uncertainty quantification, and explicit accounting for variance as essential components of reliable LLM evaluation.

Closely Related Variance-Aware Benchmarks. Only a few recent efforts go beyond calls to action and directly propose frameworks for variance-aware evaluation. (Liu et al., 2024) introduces the G -Pass@k_τ metric to capture stability in reasoning tasks, though it condenses variability into a single scalar. (Madaan et al., 2024) studies variance from a different angle, analyzing differences across training seeds and checkpoints rather than stochastic decoding. (Ye et al., 2024) integrates uncertainty measures into multi-task benchmarking, showing that accuracy and certainty do not

necessarily correlate. (Wang et al., 2025) derives theoretical sample complexity bounds to support statistically sound evaluations at lower cost. Autonomous or domain-specific benchmarks such as (Karia et al., 2024) and (Ji et al., 2025) highlight the growing recognition of reliability in evaluation, though they do not systematically address run-to-run variance.

Our Work. Our work builds on this trajectory by making stability across multiple, independent runs as the central object of our study. We echo the call to action for reliable benchmarking and reproducible science and claim that an important additional analysis is the sampling budget. We find that modern reasoning algorithms may reach state-of-the-art accuracy but only at a disproportionate cost. At the same time, the most sophisticated algorithms also seem to be the most brittle. The question of sample efficiency is closely related to reliable accuracy and reproducible results.

While prior efforts either stress brittleness under perturbations or argue for statistical rigor, REASONBENCH is, to our knowledge, the first benchmark that systematically quantifies stability across reasoning strategies, models, and tasks through controlled multi-run evaluation. By coupling reproducible implementations of reasoning strategies with a variance-aware analysis, we aim to make stability and reliability first-class metrics in LLM reasoning research.

B Additional Experimental Details

B.1 Detailed Task Descriptions

B.1.1 Game of 24

The Game of 24 is a math puzzle where players are given four numbers and must use each of them exactly once, along with the basic arithmetic operations (+, –, ×, ÷), to form an expression that evaluates to 24.

Our benchmark includes 1,362 such puzzles collected from 4nums.com, organized in ascending order of difficulty. Each puzzle provides four input numbers, and the goal is to generate a valid equation that results in 24. Following the approach of ToT Yao et al. (2023a), we designate puzzles numbered 901 to 1000 as our test set.

B.1.2 SciBench

SciBench Wang et al. (2024a) is a scientific reasoning benchmark designed to evaluate college-level problem-solving abilities across subjects such as mathematics, physics, and chemistry. Each task presents an open-ended problem that requires multi-step reasoning, domain-specific knowledge, and advanced computations, including calculus and differential equations. Problems are drawn from widely used textbooks and university exams.

Following the approach of ReST-MCTS Zhang et al. (2024), we sampled 109 problems spanning different subjects to form the test set. Quality is measured using an *accuracy* metric, defined as the proportion of problems correctly solved according to the official solutions (exact matching).

B.1.3 HumanEval

HumanEval Chen et al. (2021) is a code generation benchmark where participants are given natural language docstrings and must generate Python functions that correctly implement the described behavior. Each problem includes a hidden test suite used to verify functional correctness.

Following the setup from Reflexion Shinn et al. (2023), the benchmark consists of 100 programming tasks in the test set. We evaluate performance using the *pass@1* metric, which measures the proportion of problems solved correctly on the first attempt.

B.1.4 HotpotQA

HotpotQA Zhilin et al. (2018) is a large-scale question answering benchmark that tests an agent's ability to perform multi-hop reasoning across multiple documents. Multi-step approaches, such as ToT, are permitted to interact with an API that enables document retrieval and targeted information lookup.

Following prior work Zhou et al. (2024a); Shinn et al. (2023), we evaluate on a set of 100 randomly selected questions. The quality of a response is judged based on *exact match* (EM) with the oracle answer.

B.1.5 SHAKESPEAREAN SONNET WRITTING

Shakespearean Sonnet Writing Suzgun & Kalai (2024) is a creative generation task where the goal is to compose a 14-line sonnet adhering to the classic rhyme scheme "ABAB CDCD EFEF GG". Each sonnet must include three provided words verbatim.

Following Suzgun et al. Suzgun & Kalai (2024), we randomly sampled 50 datapoints to form the test set. Quality is measured using an *accuracy* metric, which reflects the proportion of sonnets that both satisfy the rhyme scheme and include all three required words exactly as given.

B.1.6 HUMANITY'S LAST EXAM

Humanity's Last Exam Phan et al. (2025) is a challenging, multidisciplinary benchmark designed to probe the upper limits of general reasoning and knowledge in large language models. The benchmark consists of carefully curated questions spanning mathematics, natural sciences, humanities, and abstract reasoning, with an emphasis on problems that require deep understanding, precise reasoning, and resistance to shallow pattern matching.

Each task is presented as a standalone question with a single correct answer, typically requiring multi-step logical inference, symbolic manipulation, or synthesis of domain knowledge. The benchmark is designed to be closed-book and does not permit external tool use.

Following the official evaluation protocol, we evaluate models on a 50 sample subset and report performance using an *accuracy* metric, defined as the proportion of questions answered exactly correctly. Correct answers are determined based on the original author's LLM as a Judge system with the recommended prompts and models (GPT-o3 Mini).

B.2 DETAILED DESCRIPTIONS OF REASONING STRATEGIES

B.2.1 INPUT-OUTPUT (IO)

A direct prompting strategy where the model maps an input to an output in a single step, without generating or exposing intermediate reasoning. IO relies entirely on the model's internal representations and is typically used as a baseline for comparison with more explicit reasoning methods.

B.2.2 CHAIN-OF-THOUGHT (COT)

Encourages the model to generate an explicit sequence of intermediate reasoning steps before producing a final answer. By verbalizing its reasoning process, CoT is expected to improve performance on multi-step and compositional reasoning tasks Wei et al. (2022).

B.2.3 CHAIN-OF-THOUGHT WITH SELF-CONSISTENCY (COT-SC)

Extends Chain-of-Thought by sampling multiple independent reasoning chains and aggregating their final answers via a consistency-based voting mechanism. This approach mitigates errors from individual reasoning paths and improves robustness and accuracy Wang et al. (2023).

B.2.4 REFLEXION

A reasoning framework that enables models to iteratively reflect on and critique their own outputs using feedback from prior attempts or environment interactions. Reflexion leverages self-evaluation to generate corrective insights, which are incorporated into subsequent reasoning steps to improve task performance over time Shinn et al. (2023).

B.2.5 TREE OF THOUGHTS (TOT)

Decomposes the problem into multiple chains of thoughts, organized in a tree structure. Thought evaluation and search traversal algorithms are utilized to solve the problem Yao et al. (2023a).

B.2.6 FLEET OF AGENTS (FOA)

Decomposes the problem into multiple chains of thoughts. Employs a genetic-type particle filtering approach to navigate through dynamic tree searches to solve the problem Klein et al. (2025).

B.2.7 GRAPH OF THOUGHTS (GOT)

Allows the organization of thoughts in a graph structure Besta et al. (2024). It introduces arbitrary graph-based thought transformations such as thought aggregation and thought refinement.

B.2.8 REACT

A reasoning method that interleaves reasoning (thought generation) and acting (taking environment-interacting actions) to solve problems interactively. Each action's output informs subsequent reasoning, enabling adaptive and dynamic problem-solving Yao et al. (2023b).

B.2.9 REASONING VIA PLANNING (RAP)

is a reasoning framework that equips Large Language Models (LLMs) with an internal world model and employs Monte Carlo Tree Search (MCTS) for strategic exploration of reasoning paths. RAP repurposes the LLM to simulate future states and evaluate potential actions, enabling deliberate planning and improved problem-solving performance Hao et al. (2023)

B.3 DETAILED REASONING MODELS DESCRIPTIONS

B.4 IMPLEMENTATION DETAILS

Platforms. GPT models were were accessed through the OpenAI API, Google models through the Gemini API while the utilization of the rest of the models was facilitated by the TogetherAI API.

Model checkpoints and prices. To compute the costs of our experiments we used the current model prices indicated by OpenAI, Gemini and Together AI, accordingly to the model. The specific models snapshot we used, along with their respective prices are presented in 3.

Table 3: Cost of each model that we have used, at this project's time of execution.

Model	Provider used	Input ($/1M)	Output ($/1M)
DeepSeek R1	Together API	3.00	7.00
Llama 4 Maverick	Together AI	0.27	0.85
GPT-4.1 mini	OpenAI API	0.40	1.60
GPT-4.1 nano	OpenAI API	0.10	0.40
Qwen3 235B Thinking	Together AI	0.65	3.00
GPT-OSS 120B	Together AI	0.15	0.60
GPT-5 mini	OpenAI API	0.25	2.00
GPT-5 nano	OpenAI API	0.05	0.40
Claude Haiku 4.5	Anthropic API	1.00	5.00
Gemini 3 Flash	Google Gemini API	0.50	3.00

B.4.1 MODEL CONFIGURATIONS

Generation parameters specified when making calls to any of the models used throughout this project. These parameters were not defined by us, but by the implementation where the respective prompts

were introduced. However, as newer models were used for this study, we only adjusted the maximum allowed completion tokens as needed to ensure compatibility and successful completion of responses.

Table 4: Generation parameters specified when making requests to a base LLM.

	max_tokens	temperature	top_p	stop
Game of 24	200	0.7	1	Null
SciBench	300	0.7	1	Null
Humanity's Last Exam	300	0.7	1	Null
HumanEval	200	0.7	1	Null
HotpotQA	300	0.7	1	Null
Sonnet Writing	800	1.0	1	Null

B.4.2 PROMPTS

To ensure a fair evaluation of the benchmarked reasoning strategies, we reuse the prompts introduced by prior methods. Whenever two strategies can utilize the same prompt, we use a shared version to enable direct comparison. For cases without existing prompts, e.g., novel reasoning strategy or base LLMs, if needed, we adapt the original prompts to facilitate the new use cases.

Due to the large number of methods and tasks presented in this paper, including all corresponding prompts would be impractical within the main text. Therefore, we provide a comprehensive collection of all prompts used in our experiments (both original and improved) on our GitHub repository: `https://anonymous.4open.science/r/ReasonBench-64B3/prompts.md`.

C ADDITIONAL RESULTS

C.0.1 THINKING EFFORT

We extend the reasoning effort study by jointly analyzing mean behavior and stability under test-time compute scaling. We evaluate GPT-5-NANO, GPT-5-MINI, and GEMINI-3 FLASH at three effort settings (*low*, *medium*, *high*) on the full ReasonBench suite. For each (model, effort) configuration, we repeat evaluation for 10 independent runs to estimate both expected performance and run-to-run variability.

We measure three quantities: (*i*) **Score** as the primary quality metric, (*ii*) **tokens_out** as a direct proxy for generation cost, and (*iii*) **Efficiency** as a quality–cost trade-off metric. For each quantity we report: **Average** performance using stratified bootstrap confidence intervals across benchmarks; **Run Deviation** (relative instability) as the typical percent deviation of runs from their benchmark-level mean; and **Global Noise** as the variance of z-scored outcomes across benchmarks, capturing benchmark-dependence after normalizing for task difficulty. The full results are summarized in Figure 4.

C.1 PER-SAMPLE SIGNIFICANCE UNDER REASONING-EFFORT SCALING

To complement aggregate averages, we analyze how reasoning effort affects outcomes at the level of individual benchmark instances. For each model, benchmark, and effort setting (*low*, *medium*, *high*), we estimate per-sample quality together with uncertainty by aggregating results over 10 independent runs and computing a confidence interval for each sample. We then compare effort levels pairwise (*low*→*medium*, *medium*→*high*, and *low*→*high*) and classify each sample that *improved*, *worsened*, or *was unnaffected* based on whether the confidence intervals indicate a statistically significant increase, a statistically significant decrease, or an overlap. Finally, we report, for each of the three models and for each benchmark, the percentage of samples in each category, quantifying how often increased effort yields meaningful gains (or regressions) beyond what is explained by run-to-run variability.

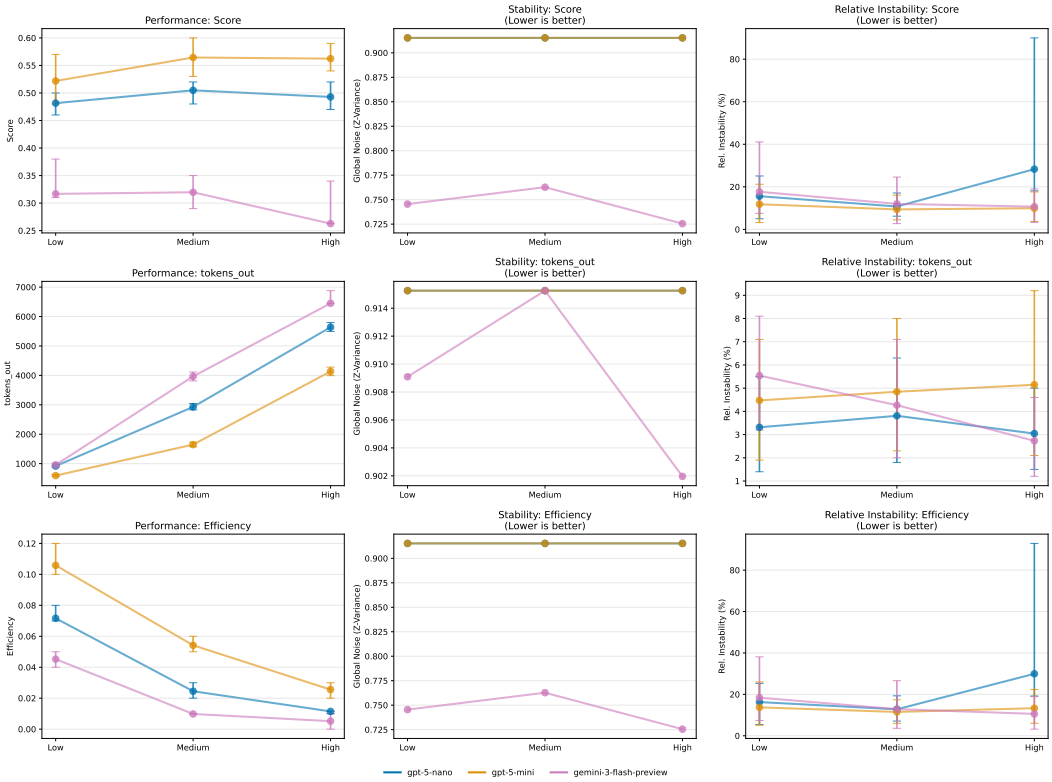


Figure 4: Effect of test-time reasoning effort on mean performance and stability for GPT-5 NANO, GPT-5 MINI, and GEMINI-3 FLASH. Rows report quality (**Score**), cost proxy (**tokens_out**), and efficiency (**Quality per token**); columns report the mean metric (**Performance**), cross-benchmark variability (**Global Noise**, lower is better), and run-to-run deviation (**Relative Instability**, lower is better). Error bars denote uncertainty estimates from repeated runs.

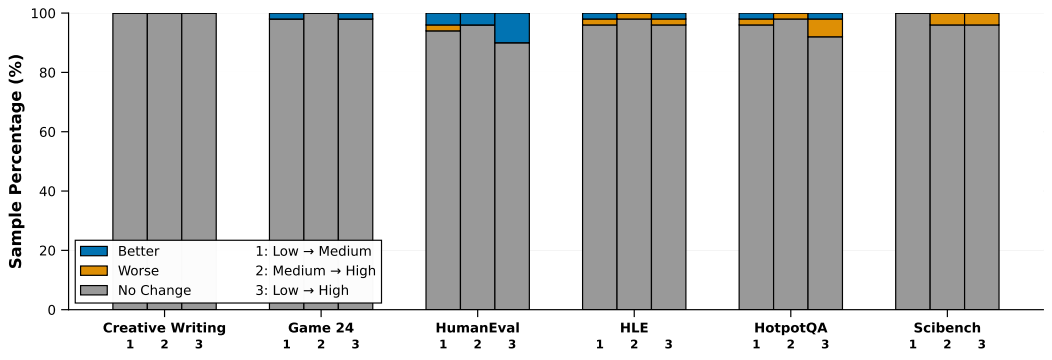


Figure 5: Per-sample significance analysis for GPT-5 Nano: for each benchmark, stacked bars show the percentage of instances whose quality is *better*, *worse*, or shows *no statistically significant change* when increasing `reasoning_effort` (low→medium, medium→high, and low→high), based on confidence-interval comparisons over 10 independent runs.

C.2 COST–QUALITY CORRELATION

Figure 8 provides per-benchmark scatter plots of sample-level cost versus quality for FoA, ReAct, and GoT, complementing the analysis in the main text.

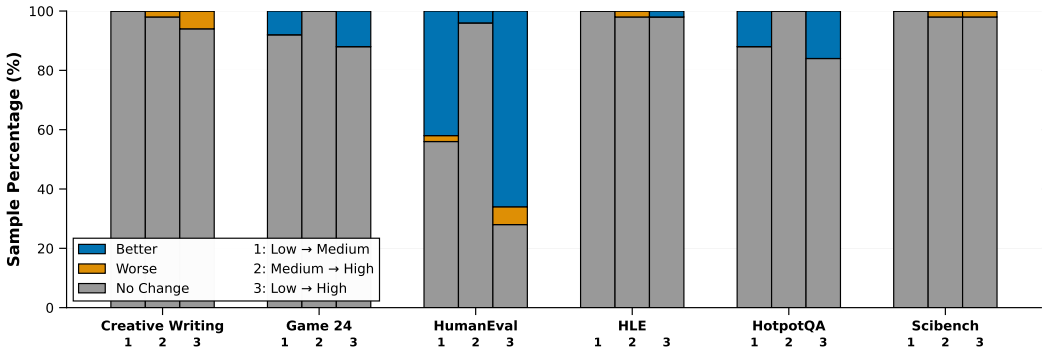


Figure 6: Per-sample significance analysis for GPT-5 Mini: for each benchmark, stacked bars show the percentage of instances whose quality is *better*, *worse*, or shows *no statistically significant change* when increasing `reasoning_effort` (low→medium, medium→high, and low→high), based on confidence-interval comparisons over 10 independent runs.

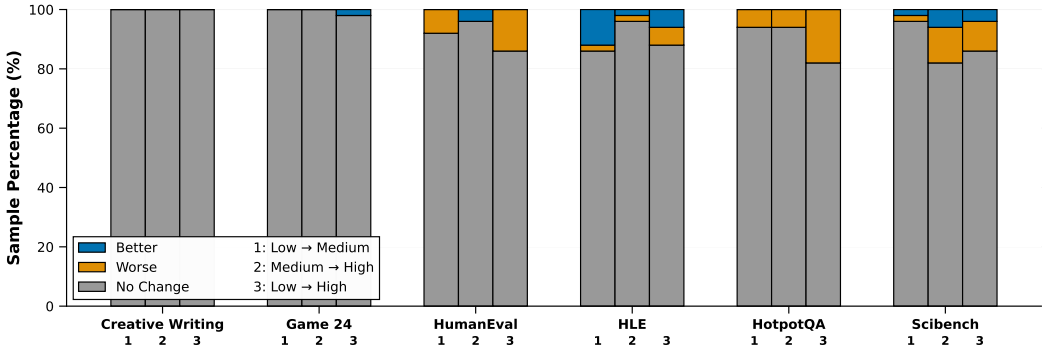


Figure 7: Per-sample significance analysis for Gemini-3 Flash: for each benchmark, stacked bars show the percentage of instances whose quality is *better*, *worse*, or shows *no statistically significant change* when increasing `reasoning_effort` (low→medium, medium→high, and low→high), based on confidence-interval comparisons over 10 independent runs.

C.3 Impact of Prompt Refinements on Strategy Performance

Table 5 reports per-strategy quality before and after prompt and parsing refinements. All strategies improve significantly, with direct methods (IO, CoT, CoT-SC) benefiting the most, while run-to-run variance remains largely unchanged.

C.3.1 Causal Analysis

In the main paper, we report the causal intervention results using mean quality to isolate the effect of replacing heuristic evaluation with ground-truth evaluation. In the appendix, we provide a fuller stability characterization in Fig, 9 by additionally reporting the variance of quality in and the relative run deviation (with confidence intervals) for the same intervention, highlighting how evaluation accuracy affects not only expected performance but also run-to-run variability.

C.3.2 Diagnosis

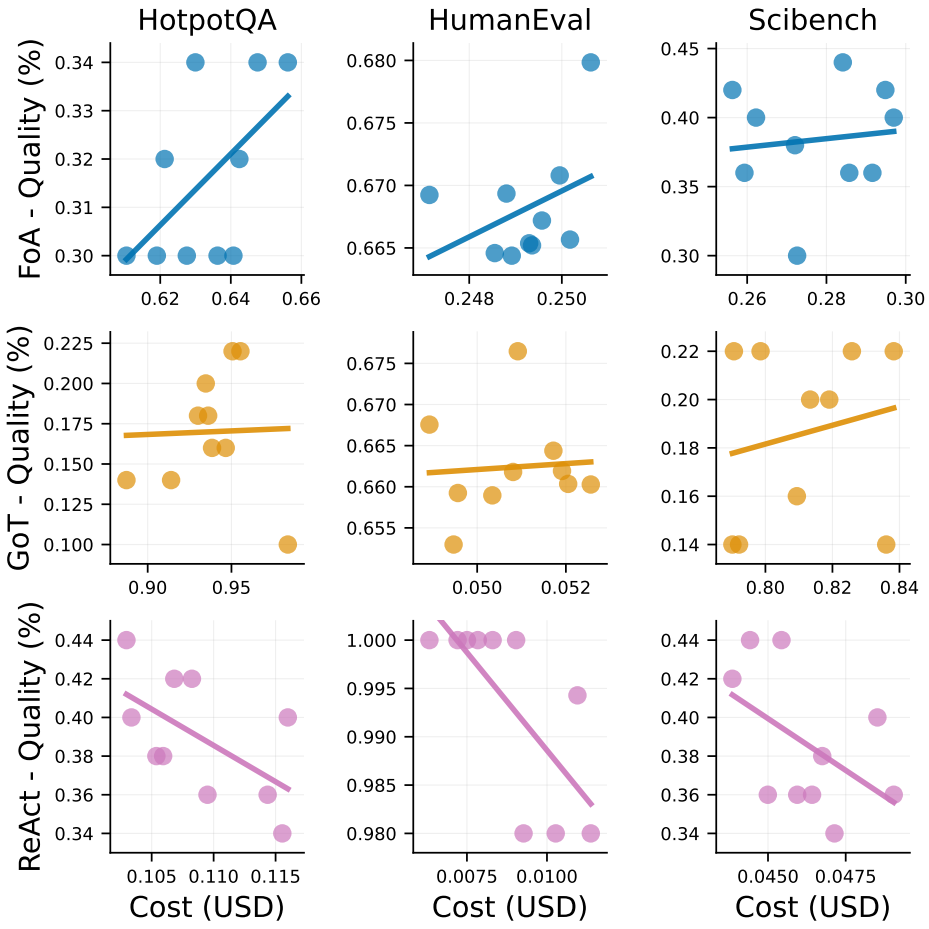


Figure 8: **Correlation between Quality and Cost.** For FoA, quality scales positively with cost across all benchmarks. ReAct exhibits a consistent negative slope, indicating diminishing returns at higher costs. GoT does not follow a uniform pattern, with its cost–quality relationship varying substantially by task.

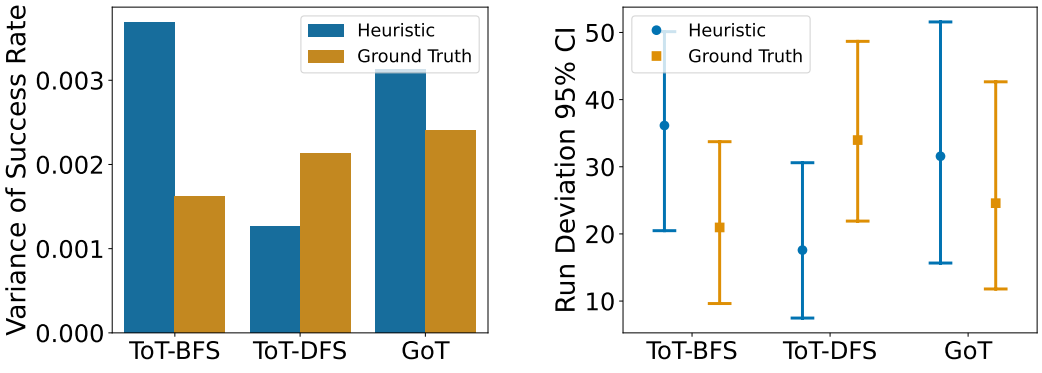


Figure 9: Run-to-run deviation (95% CI) and variance of solution quality under heuristic versus ground-truth evaluation for ToT-BFS, ToT-DFS, and GoT on *Game of 24*.

Table 5: **Impact of prompt and parsing refinements on strategy performance.** Enhancing clarity and standardizing output parsing significantly improves accuracy without affecting the stability. Direct prompting methods show the largest gains while the rest showcase similar ones, except RAP which improves the least. The best performance is shown in **blue** whereas the worst is shown in **orange**.

Strategy	Type	Original Prompts*	Improved Prompts*	Δ
IO	Direct	**0.106 [0.10, 0.12]**	0.313 [0.28, 0.34]	**+0.207** †
CoT	Direct	0.276 [0.25, 0.30]	0.398 [0.35, 0.43]	+0.122†
CoT-SC	Direct	0.228 [0.21, 0.24]	0.410 [0.40, 0.45]	+0.182†
ReAct	Adaptive	0.295 [0.28, 0.31]	0.391 [0.36, 0.42]	+0.096†
Reflexion	Adaptive	0.282 [0.27, 0.30]	0.411 [0.39, 0.42]	+0.129†
ToT-DFS	Structured	0.127 [0.10, 0.14]	**0.177 [0.15, 0.20]**	+0.050†
GoT	Structured	0.3361 [0.31, 0.36]	0.420 [0.39, 0.46]	+0.084†
ToT-BFS	Structured	0.407 [0.38, 0.44]	0.506 [0.47, 0.54]	+0.099†
RAP	Planning	0.367 [0.35, 0.38]	0.403 [0.39, 0.41]	**+0.036** †
FoA	Evolutionary	**0.4580 [0.43, 0.48]**	**0.546 [0.52, 0.58]**	+0.088†

† Indicates statistical significance ($p < 0.05$) from original.
* Reports average quality and 95% confidence intervals in brackets.

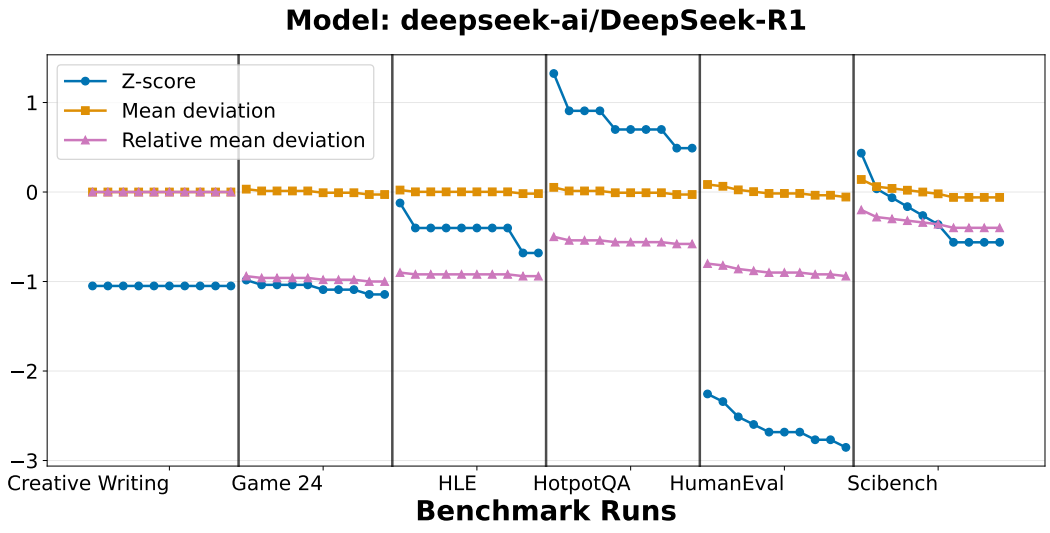


Figure 10: Run-level stability trace: per-run outcomes grouped by benchmark, showing benchmark-normalized z-scores and absolute/relative deviations from the benchmark mean across repeated runs.

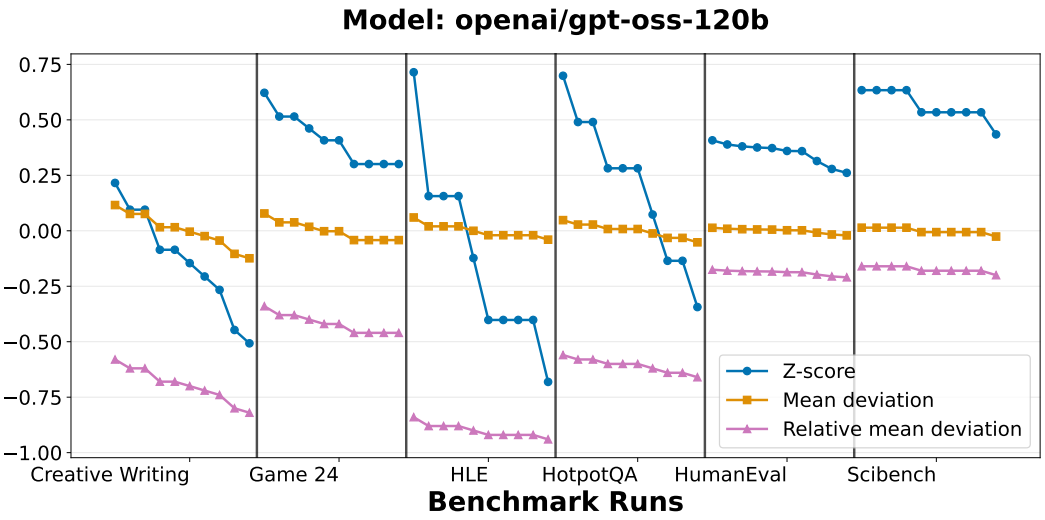


Figure 11: Run-level stability trace for DeepSeek-R1: per-run outcomes grouped by benchmark, showing benchmark-normalized z-scores and absolute/relative deviations from the benchmark mean across repeated runs.

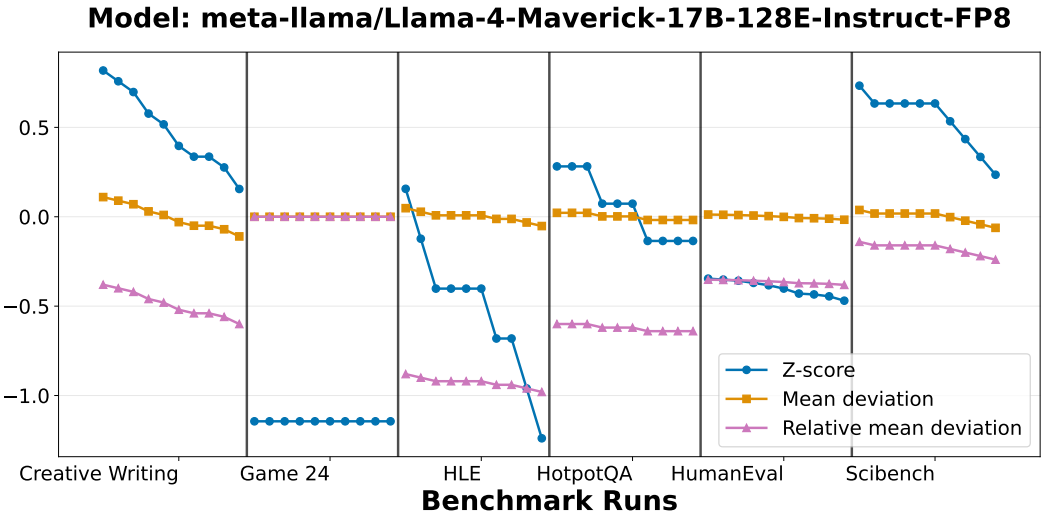


Figure 12: Run-level stability trace for Llama-4-Maverick: per-run outcomes grouped by benchmark, showing benchmark-normalized z-scores and absolute/relative deviations from the benchmark mean across repeated runs.

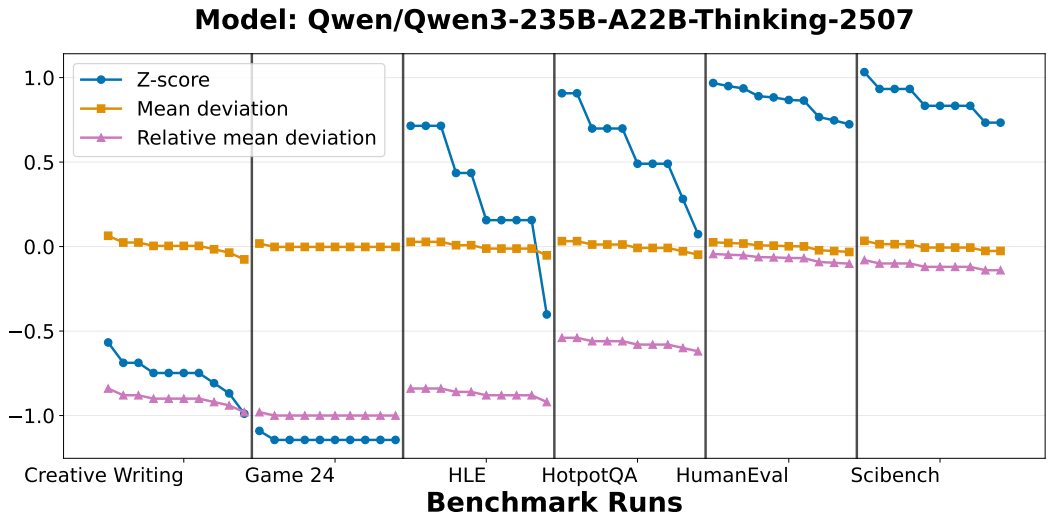


Figure 13: Run-level stability trace for Qwen3-235B Thinking: per-run outcomes grouped by benchmark, showing benchmark-normalized z-scores and absolute/relative deviations from the benchmark mean across repeated runs.

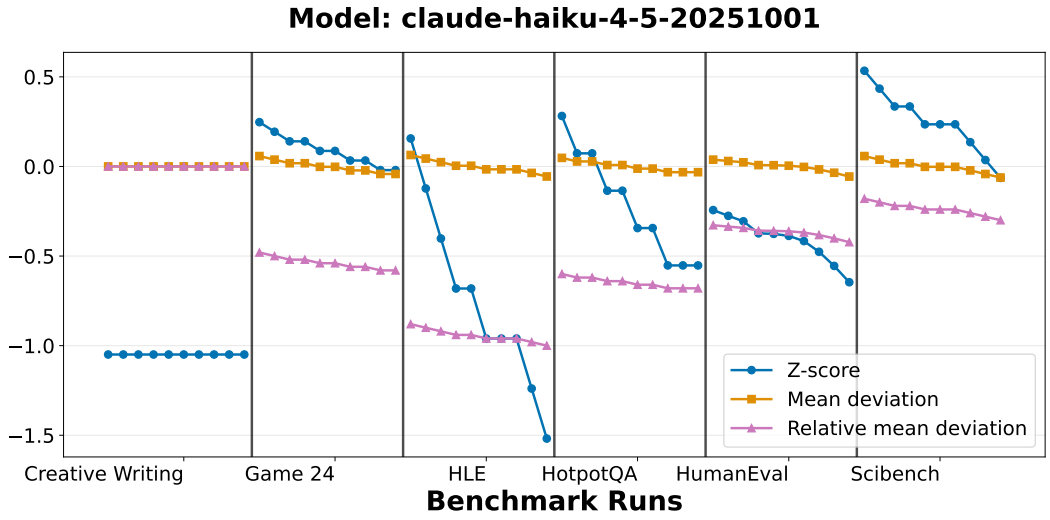


Figure 14: Run-level stability trace for Claud-Haiku 4.5: per-run outcomes grouped by benchmark, showing benchmark-normalized z-scores and absolute/relative deviations from the benchmark mean across repeated runs.

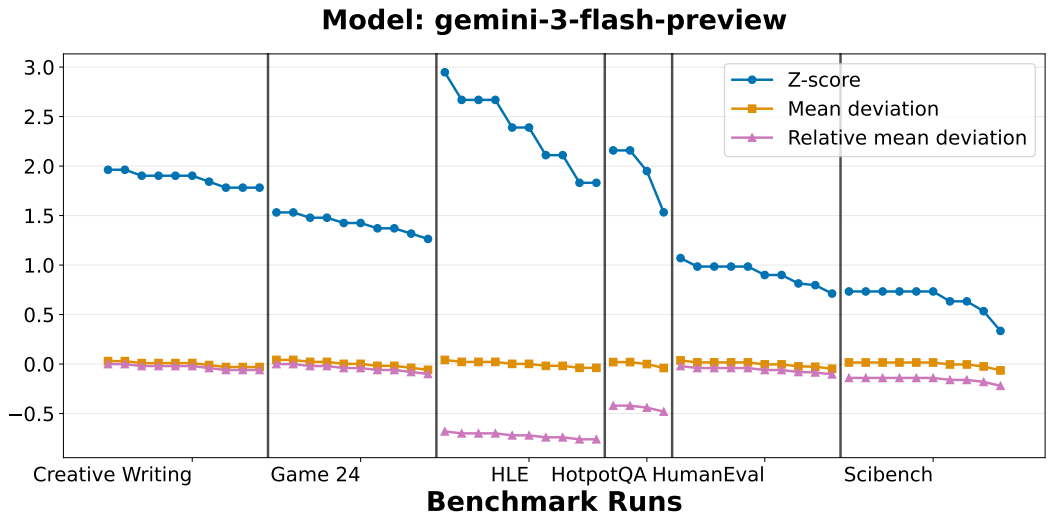


Figure 15: Run-level stability trace for Gemini-3 Flash: per-run outcomes grouped by benchmark, showing benchmark-normalized z-scores and absolute/relative deviations from the benchmark mean across repeated runs.

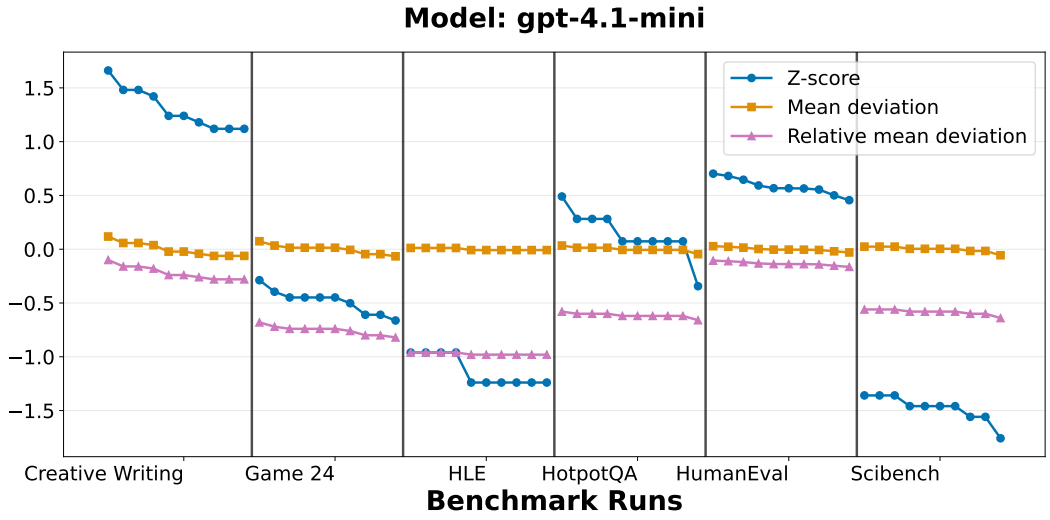


Figure 16: Run-level stability trace for GPT-4.1 Mini: per-run outcomes grouped by benchmark, showing benchmark-normalized z-scores and absolute/relative deviations from the benchmark mean across repeated runs.

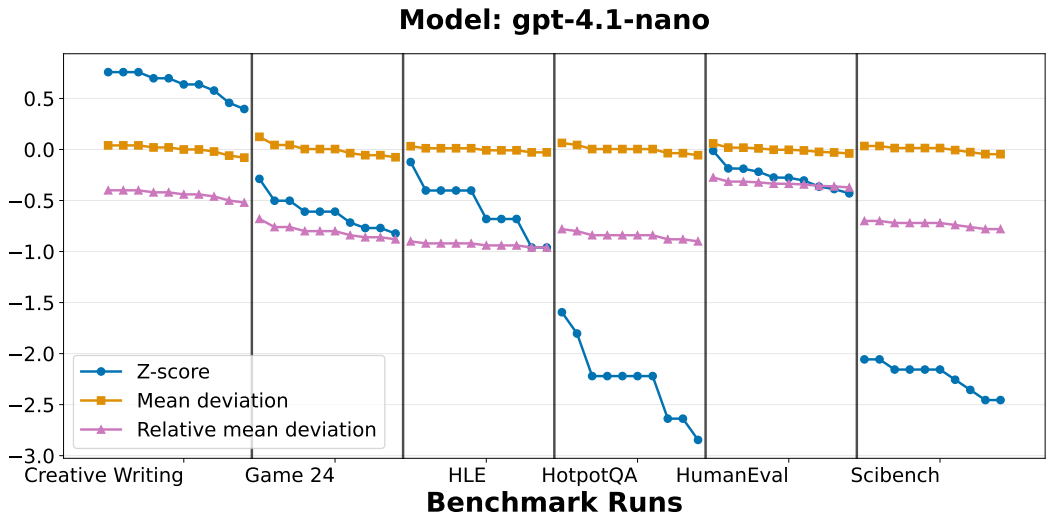


Figure 17: Run-level stability trace for GPT-4.1 Nano: per-run outcomes grouped by benchmark, showing benchmark-normalized z-scores and absolute/relative deviations from the benchmark mean across repeated runs.

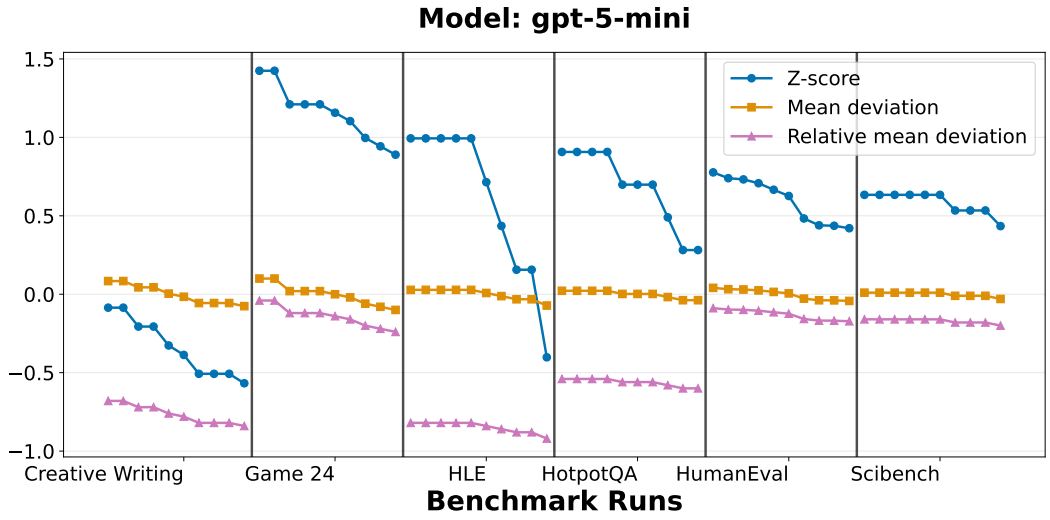


Figure 18: Run-level stability trace for GPT-5 Mini: per-run outcomes grouped by benchmark, showing benchmark-normalized z-scores and absolute/relative deviations from the benchmark mean across repeated runs.

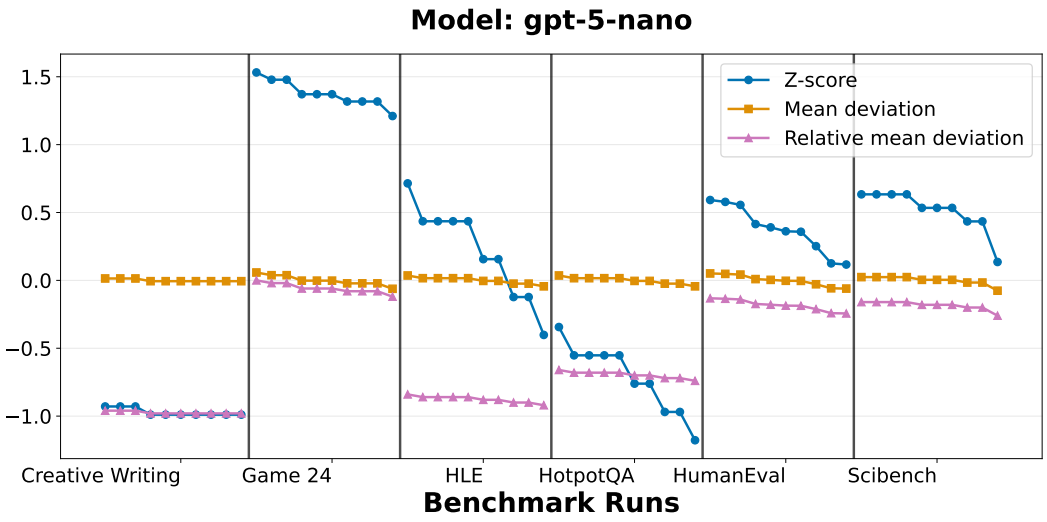


Figure 19: Run-level stability trace for GPT-5 Nano: per-run outcomes grouped by benchmark, showing benchmark-normalized z-scores and absolute/relative deviations from the benchmark mean across repeated runs.