Silicon Bureaucracy and AI Test-Oriented Education: Contamination Sensitivity and Score Confidence in LLM Benchmarks

Yiliang Song[1,2†], **Hongjun An**[1,3†], **Jiangan Chen**[2], **Xuanchen Yan**[3], **Huan Song**[1], **Jiawei Shao**[1], **Xuelong Li**[1*]

[1] Institute of Artificial Intelligence (TeleAI), China Telecom
[2] Guangxi Normal University [3] Northwestern Polytechnical University
[†] Equal contribution; work done while interning at TeleAI.
[*] Correspondence to Xuelong Li <xuelong_li@ieee.org>.

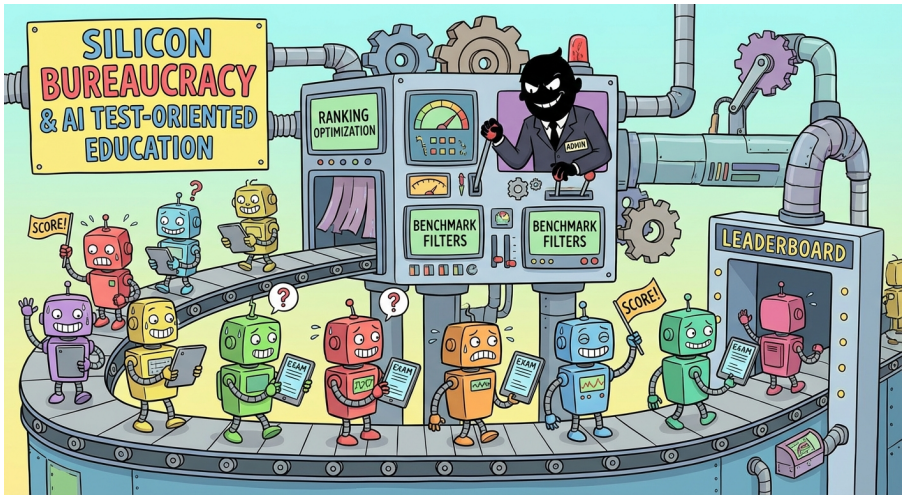

Figure 1: Silicon Bureaucracy and AI Test-Oriented Education.

Abstract

Public benchmarks increasingly govern how large language models (LLMs) are ranked, selected, and deployed. We frame this benchmark-centered regime as Silicon Bureaucracy and AI Test-Oriented Education, and argue that it rests on a fragile assumption: that benchmark scores directly reflect genuine generalization. In practice, however, such scores may conflate exam-oriented competence with principled capability, especially when contamination and semantic leakage are difficult to exclude from modern training pipelines. We therefore propose an audit framework for analyzing contamination sensitivity and score confidence in LLM benchmarks. Using a router-worker setup, we compare a clean-control condition with noisy conditions in which benchmark problems are systematically deleted, rewritten, and perturbed before being passed downstream. For a genuinely clean benchmark, noisy conditions should not consistently outperform the clean-control baseline. Yet across multiple models, we find widespread but heterogeneous above-baseline gains under noisy conditions, indicating that benchmark-related cues may be reassembled and can reactivate contamination-related memory. These results suggest that similar benchmark scores may carry substantially different levels of

confidence. Rather than rejecting benchmarks altogether, we argue that benchmark-based evaluation should be supplemented with explicit audits of contamination sensitivity and score confidence.

1 Introduction

The development of large language models (LLMs) is increasingly organized around the scores, rankings, and leaderboards produced by public benchmarks [Kwan et al., 2024, Bai et al., 2024, Sun et al., 2024, Chen et al., 2024, An et al., 2026a, Shao and Li, 2025, Song et al., 2025, 2026a, Yuan et al., 2025]. In academia, industry, and the broader public sphere, benchmark scores are no longer merely technical indicators for research communication [Alzahrani et al., 2024, Wang et al., 2026, Johri et al., 2025]. They have gradually become evaluative criteria with real institutional consequences: models are examined, ranked, filtered, and treated as reference objects in procurement, investment, and governance decisions [Alzahrani et al., 2024, Pham, 2025, Bean et al., 2025, Wang et al., 2026]. In this sense, benchmarks have shifted from research tools to institutionalized examination and selection devices [Pham, 2025, Bean et al., 2025, Center for AI Safety et al., 2026]. The ranking, certification, and comparison logic built around them reflects a pattern that deserves critical reflection, namely, Silicon Bureaucracy and AI Test-Oriented Education.

Yet this institutionalized mode of evaluation rests on a strong assumption: a high benchmark score reliably indicates stronger and more genuine generalization ability [Bean et al., 2025, Huang et al., 2025a, Wang et al., 2026]. This paper argues that the assumption is not robust [Alzahrani et al., 2024, Bean et al., 2025, Sun et al., 2025]. Current LLM benchmarks often measure not one pure notion of "true generalization," but a mixture of two qualitatively different capacities [Bean et al., 2025, Pham, 2025]. One is exam-oriented competence: under fixed answering formats, input–output conventions, and judging rules, the model can produce the correct answer [Pham, 2025, Alzahrani et al., 2024]. The other is the ability that remains after contamination and semantic leakage are excluded as much as possible, namely principled understanding and transfer to genuinely unseen tasks [Zhang et al., 2024b, Huang et al., 2025a, Sun et al., 2025]. The former is closer to "can it answer," while the latter is closer to "can it truly generalize" [Bean et al., 2025, Huang et al., 2025a]. In practice, however, many benchmarks compress both into a single score and then treat that score as a proxy for overall capability [Alzahrani et al., 2024, Bean et al., 2025, Zhang et al., 2024a, Liu et al., 2023, Johri et al., 2025].

A central reason why this interpretation is unstable is that benchmark-related information is extremely difficult to remove from the training pipeline [Dekoninck et al., 2024, Ni et al., 2025, Sun et al., 2025]. LLM training data are massive and heterogeneous [Ni et al., 2025, Sun et al., 2025], and after repeated crawling, cleaning, distillation, synthesis, and alignment, it is hard to guarantee that benchmark questions, answers, or closely related variants never entered training [Dekoninck et al., 2024, Ni et al., 2025]. Exact deduplication may still miss paraphrases, solution fragments, discussions, distilled samples, or synthetic variants that remain semantically close to the originals [Dekoninck et al., 2024, Sun et al., 2025]. Post-training blocking of original questions or keywords may also fail to prevent semantically neighboring inputs or the indirect activation created by aggregating partial clues [Dekoninck et al., 2024, Ni et al., 2025]. The benchmark problem, therefore, is not only whether exact test items appeared in training, but also whether the model encountered generalized information sufficient to point toward the correct answer [Dekoninck et al., 2024, Sun et al., 2025]. Moreover, because leaderboard competition, promotion, and selection increasingly depend on benchmark performance, model development may contain latent incentives to optimize for exam success [Alzahrani et al., 2024, Pham, 2025]. This further blurs the boundary between optimizing for benchmark performance and optimizing for genuine generalization [Pham, 2025, Bean et al., 2025]. As a result, a high score may reflect not only stronger true generalization, but also stronger exploitation of contamination-related signals [Dekoninck et al., 2024, Sun et al., 2025].

Motivated by these concerns, this paper proposes an audit framework for interpreting benchmark scores and assessing their credibility [Song et al., 2026b, Bouchard et al., 2026, Sokol et al., 2024, Wang et al., 2026]. Rather than asking only which model scores higher, we ask whether performance shows an anomalous pattern when the information in a problem is systematically deleted, rewritten, and perturbed with noise. To study this, we model a single system as upstream routers and a downstream worker. Under a clean control condition, routers transmit the original problem as completely as possible; under noisy conditions, they delete, rewrite, and perturb it, and the aggregated outputs

are then sent to the worker. If a benchmark is genuinely clean, performance under noisy conditions should at most approach the clean-control baseline, but should not persistently or systematically exceed it. Once a stable above-baseline phenomenon appears, a more plausible explanation is not that noise makes the model stronger, but that deleted, rewritten, and aggregated information has reassembled into cues capable of reactivating contamination-related memory traces [Dekoninck et al., 2024, Sun et al., 2025]. In this way, we transform the question of "what benchmark scores actually measure" into an audit problem that can be computed, compared, and used for model evaluation and selection [Song et al., 2026b, Bouchard et al., 2026, Wang et al., 2026].

The contributions of this paper are threefold. First, we reinterpret LLM benchmarks from the perspective of institutionalized examination and selection, and propose the framework of Silicon Bureaucracy and AI Test-Oriented Education. Second, we introduce a router–worker-based audit method that identifies sensitivity to potential contamination cues by comparing deviations between clean-control and noisy conditions, especially above-baseline gains that should not systematically occur in theory. Third, across multiple models, we show that such anomalous gains are widespread but heterogeneous, implying that even similar benchmark scores may differ substantially in credibility. Accordingly, we do not claim that benchmarks are entirely invalid; rather, we argue that benchmark scores in the LLM era should be reinterpreted and supplemented with explicit credibility auditing.

2 Related Work

The problem addressed in this paper lies at the intersection of three lines of research: one focuses on LLM benchmarks and leaderboard-based evaluation, another on data contamination, deduplication, and benchmark leakage, and the third on model stability and selectability under different protocols and interaction conditions [Alzahrani et al., 2024, Dekoninck et al., 2024, Song et al., 2026b, Zhu et al., 2024]. Relative to these lines of work, our goal is not simply to identify whether a particular benchmark has been leaked, nor to revisit the familiar question of which model has higher average performance. Rather, we seek to reconsider the meaning of benchmark scores themselves: once benchmarks have evolved into institutionalized examination and selection devices, what exactly do these scores measure, and how credible are they as indicators of genuine generalization ability [Bean et al., 2025, Wang et al., 2026, Bouchard et al., 2026]?

2.1 LLM Benchmarks and Leaderboard-Based Evaluation

As competition over LLM capability intensifies, benchmarks have gradually evolved from shared measurement tools among researchers into ranking tools, publicity tools, and practical bases for deployment decisions [Kwan et al., 2024, Bai et al., 2024, Ye et al., 2025, Huang et al., 2025b, Zhao et al., 2025, Sun et al., 2024, Chen et al., 2024, Center for AI Safety et al., 2026]. Whether a model enters the top tier of a leaderboard affects not only academic reputation, but also product comparison, user perception, investment judgment, and actual procurement decisions [Alzahrani et al., 2024, Wang et al., 2026, Johri et al., 2025]. In this process, benchmarks are no longer merely neutral technical measuring instruments; they increasingly take on institutional functions [Pham, 2025, Bean et al., 2025, Center for AI Safety et al., 2026]. Scores resemble résumés, leaderboards resemble performance review tables, and high-scoring models are more likely to obtain the status of being seen as "advanced," "reliable," or "deployable" [Alzahrani et al., 2024, Wang et al., 2026]. Existing research and practice have paid more attention to how benchmarks can be used to quickly compare models, but have paid less attention to whether the scores on which such institutionalized comparisons rely are themselves equally credible [Li et al., 2025, Wang et al., 2026, Sokol et al., 2024, Bean et al., 2025]. It is precisely at this point that this paper takes a further step: our concern is not whether benchmarks are useful, but how benchmark scores should be reinterpreted once benchmarks have become institutionalized examination devices [Pham, 2025, Bean et al., 2025].

2.2 Data Contamination, Deduplication, and Benchmark Leakage

Research on benchmark contamination has mainly discussed exact question leakage, near-duplicates, train–test overlap, and the evaluation biases that follow from them [Dekoninck et al., 2024, Ni et al., 2025, Sun et al., 2025]. Related work usually emphasizes deduplication, filtering, and cleaning of training corpora as important means of preventing benchmark contamination [Dekoninck et al., 2024, Ni et al., 2025]. However, this paper argues that exact deduplication does not imply the disappearance

of contamination [Sun et al., 2025, Dekoninck et al., 2024]. Even if the original benchmark questions themselves are removed, paraphrased texts that are semantically close to the originals, solution fragments, discussion records, distilled samples, or synthetic data may still remain in the training pipeline in the form of generalized information, and may still indirectly point toward the correct answer at evaluation time [Dekoninck et al., 2024, Ni et al., 2025, Sun et al., 2025]. Furthermore, if post-training interventions only attempt to block original questions, reference answers, or keywords, they may still fail to block rewritten inputs or to prevent the reactivation of related memory traces after multiple partial clues are aggregated together [Dekoninck et al., 2024, Ni et al., 2025]. Therefore, the benchmark problem should not be understood merely as whether the exact original questions entered the training set; it should also be understood as whether the model has already encountered semantically neighboring information sufficient to point toward the correct answer [Dekoninck et al., 2024, Sun et al., 2025]. What this paper emphasizes is precisely this broader and harder-to-govern form of contamination, which extends beyond exact question leakage [Dekoninck et al., 2024, Sun et al., 2025].

2.3 Stability Evaluation, CreditAudit, and the Interpretation of Scores

Beyond contamination research, recent work has also begun to recognize that model selection cannot rely on a single average score alone, but must also consider model stability across different interaction protocols, prompt templates, and task organizations [Song et al., 2026b, Zhu et al., 2024, Bai et al., 2024, An et al., 2026b]. Research represented by CreditAudit[Song et al., 2026b] argues that model evaluation in engineering settings is not only about which model has higher average ability, but also about which model remains more stable under institutionalized calling conditions. This perspective directly informs the present paper. The difference is that CreditAudit is more concerned with a model's sensitivity to protocol and scenario variation, that is, protocol sensitivity, whereas this paper further asks about the sensitivity of benchmark scores to potential contamination cues, that is, contamination sensitivity [Song et al., 2026b, An et al., 2026b]. Put differently, the former is more concerned with whether a model is stable, whereas this paper is further concerned with whether its benchmark score is trustworthy [Song et al., 2026b, Bouchard et al., 2026]. Even when two models obtain similar benchmark scores, those scores do not necessarily have the same degree of credibility [Song et al., 2026b, An et al., 2026b, Bouchard et al., 2026]. Accordingly, model evaluation should not stop at comparing score levels alone; it should also ask to what extent those scores may have been influenced by contamination-related cues [Song et al., 2026b, Dekoninck et al., 2024, Bouchard et al., 2026].

3 Methodology and Hypotheses

This section formalizes benchmark scores as empirical performance under an observed distribution and studies the conditions under which such scores can be interpreted as indicators of genuine generalization. We define benchmark score, score confidence, and contamination sensitivity; introduce a router–worker mechanism; and derive the theoretical judgment that, in contamination-free settings, noisy aggregation should not systematically outperform the clean baseline.

3.1 Conceptual Framework

Let $\mathcal{Q}$ denote the question space, $\mathcal{A}$ the answer space, and $\mathcal{B}_n = \{(q_i, a_i)\}_{i=1}^n \subset \mathcal{Q} \times \mathcal{A}$ a benchmark sample. For model θ, define

$$Y_i(\theta) = \mathbf{1}\{\hat{a}_\theta(q_i) = a_i\}, \qquad \hat{s}_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} Y_i(\theta). \tag{1}$$

and let the corresponding population score under the benchmark distribution P_B be

$$s_B(\theta) = \int_{\mathcal{Q}\times\mathcal{A}} \mathbf{1}\{\hat{a}_\theta(q) = a\}\, dP_B(q, a). \tag{2}$$

Let P_0 denote an idealized target distribution under which benchmark-related contamination has been excluded as much as possible. Then the model's contamination-reduced ability is

$$s_0(\theta) = \int_{\mathcal{Q}\times\mathcal{A}} \mathbf{1}\{\hat{a}_\theta(q) = a\}\, dP_0(q, a), \qquad \Delta(\theta) = s_B(\theta) - s_0(\theta). \tag{3}$$

We define score confidence as the credibility of the benchmark score as an indicator of genuine generalization ability, written abstractly as

$$\mathrm{Conf}(\theta) = \phi(|\Delta(\theta)|), \qquad \phi'(\cdot) < 0. \tag{4}$$

Thus, a higher score does not necessarily imply a higher-confidence score.

We further define contamination sensitivity as the responsiveness of model performance to contamination-related cues. Let $\lambda \geq 0$ denote cue intensity and $m(\theta, \lambda) = \mathbb{E}[Y(\theta; \lambda)]$ the expected correctness rate. Then

$$\mathrm{CS}(\theta) = \left. \frac{\partial m(\theta, \lambda)}{\partial \lambda} \right|_{\lambda = 0^+}. \tag{5}$$

A larger value indicates that the benchmark score is more likely to contain non-generalization components.

3.2 Router–Worker Mechanism

For question $q \in \mathcal{Q}$, let $S(q)$ denote its latent task-relevant information. Under the clean-baseline condition, a single router $R^c_\theta : \mathcal{Q} \to \mathcal{Z}$ transmits the problem as completely as possible, producing $Z^c = R^c_\theta(q)$. The worker $W_\theta : \mathcal{Z} \to \mathcal{A}$ then answers on the basis of router output only:

$$\hat{a}^c_\theta(q) = W_\theta(R^c_\theta(q)), \qquad Y^c(q, a; \theta) = \mathbf{1}\{\hat{a}^c_\theta(q) = a\}, \qquad \hat{s}^c_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} Y^c(q_i, a_i; \theta). \tag{6}$$

Under noisy conditions, there are m parallel routers $R^n_{\theta,1}, \ldots, R^n_{\theta,m}$. Each router deletes, rewrites, and perturbs the original problem, producing $Z^n_j = R^n_{\theta,j}(q)$. Their outputs are aggregated by $A_m : \mathcal{Z}^m \to \mathcal{T}$ into

$$T_m = A_m(Z^n_1, \ldots, Z^n_m), \tag{7}$$

and the worker answers only on the basis of T_m:

$$\hat{a}^{n,m}_\theta(q) = W_\theta(T_m), \qquad Y^{n,m}(q, a; \theta) = \mathbf{1}\{\hat{a}^{n,m}_\theta(q) = a\}, \qquad \hat{s}^{n,m}_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} Y^{n,m}(q_i, a_i; \theta). \tag{8}$$

The score deviation of the noisy condition relative to the clean baseline is

$$G_m(\theta) = \hat{s}^{n,m}_n(\theta) - \hat{s}^c_n(\theta), \qquad G^+_m(\theta) = \max\{G_m(\theta), 0\}. \tag{9}$$

Whenever $G_m(\theta) > 0$, the model is said to exhibit an above-baseline anomaly under router count m.

3.3 Theoretical Judgment

Let $N(q)$ denote information unrelated to the correct answer. A noisy router output can be abstractly written as

$$Z^n_j = D_j(S(q)) \cup U_j, \qquad D_j(S(q)) \subseteq S(q), \quad U_j \subseteq N(q) \cup \tilde{N}_j, \tag{10}$$

where $D_j(\cdot)$ is a deletion operator and $\tilde{N}_j$ denotes exogenous perturbation. Hence the aggregated input received by the worker is

$$T_m \sim A_m(D_1(S(q)) \cup U_1, \ldots, D_m(S(q)) \cup U_m). \tag{11}$$

Under contamination-free conditions, let the worker's success probability be $\pi_\theta(T) = \mathbb{P}(W_\theta(T) = a \mid q, a)$, assumed to be weakly increasing in effective information and weakly decreasing in irrelevant noise. Then noisy aggregation can improve performance only through cross-router complementarity, while also introducing extra noise. This yields

$$\mathbb{E}[Y^{n,m}(q, a; \theta) \mid q, a] \leq \mathbb{E}[Y^c(q, a; \theta) \mid q, a] + \varepsilon_m(q, \theta). \tag{12}$$

and thus

$$\mathbb{E}[G_m(\theta)] \leq \bar{\varepsilon}_m(\theta), \qquad \bar{\varepsilon}_m(\theta) = \int \varepsilon_m(q, \theta)\, dP_B(q, a). \tag{13}$$

If the clean baseline is already close to full-information transmission, $\bar{\varepsilon}_m(\theta)$ should be small. In that case, noisy conditions may fluctuate around the baseline, but they should not systematically exceed it.

Now let $\Xi(q)$ denote a latent set of benchmark-related contamination cues, including paraphrased variants, solution fragments, discussion texts, distilled samples, synthetic variants, or semantically neighboring expressions that remain reachable even after local post-training blocking. If the overlap between aggregated text and contamination cues is measured by $\kappa(T_m, \Xi(q))$, then the worker's success probability may be written as

$$\pi_\theta(T_m, \Xi(q)) = \pi_\theta^0(T_m) + \psi_\theta(\kappa(T_m, \Xi(q))), \qquad \psi'_\theta(\cdot) \geq 0. \tag{14}$$

As m increases, partial clues from different noisy routers may accumulate, increasing $\kappa(T_m, \Xi(q))$. Then

$$\mathbb{E}[Y^{n,m}(q, a; \theta) \mid q, a] > \mathbb{E}[Y^c(q, a; \theta) \mid q, a] \tag{15}$$

may hold for a nontrivial subset of questions, so that persistent positive $G_m(\theta)$ is more naturally interpreted as an external signal of contamination-related memory activation than as an ordinary noise effect.

3.4 Hypotheses

H1. Contamination activation hypothesis. If benchmark-related semantic-neighbor contamination exists during training, or if post-training interventions only block original benchmark items locally, then multi-router noisy aggregation is more likely to activate generalized memory related to the benchmark, thereby generating above-baseline anomalies under some router settings.

H2. Heterogeneous sensitivity hypothesis. Different models differ in their sensitivity to potential contamination cues. As a result, even when their benchmark scores are similar, the magnitude of anomalous gains and the breadth of violations may differ substantially.

H3. Directional transition hypothesis. If above-baseline anomalies are driven by contamination-related memory activation rather than random fluctuation, then as the number of noisy routers increases, wrong-to-correct transitions should rise, correct-to-wrong transitions should decline correspondingly, and the former should eventually exceed the latter.

4 Experimental Design

This section explains how the theoretical framework is translated into a reproducible and interpretable audit procedure. The emphasis is not on engineering complexity, but on constructing a clear comparison regime: under the same questions, the same models, and the same answering constraints, we compare the clean baseline with noisy conditions and use the resulting deviations to assess how sensitive benchmark scores are to potential contamination-related cues.

4.1 Dataset and Sample Construction

We conduct the experiments on a public benchmark consisting of multiple-choice test questions. From the test split, we draw a fixed sample of $n = 100$ questions, with the random seed set to 42. All models and all clean/noisy conditions are evaluated on exactly the same question set. This design removes additional variation caused by sampling differences across runs. In other words, the experiment follows a same-question matched comparison rather than a comparison across different question sets, so that the observed score deviations can be more directly attributed to changes in the information transmission regime.

4.2 Models and Experimental Settings

The audit is repeated across multiple mainstream large language models. By default, the router and the worker are instantiated with the same model, so that model heterogeneity does not enter the transmission process itself. The clean baseline is defined as the setting `forward_full` with $r = 1$, that is, a single router is used and is instructed to transmit the original problem information as completely as possible. The noisy conditions are defined as the setting `noisy_rewrite` with $r \in \{1, 2, \ldots, M\}$, where r denotes the number of parallel noisy routers. The role of router count is

not to test whether more agents make a model stronger; rather, it serves as a control over the intensity of cue aggregation. As r increases, more locally deleted, rewritten, and perturbed versions of the same problem are aggregated together, making it more likely that benchmark-related semantic-neighbor cues are reassembled in the final input to the worker.

4.3 Prompting and Answering Constraints

Under the clean-baseline condition, the router is instructed to preserve the original problem as fully and accurately as possible, including the question stem, options, and relevant constraints, while not directly outputting the final answer. Under noisy conditions, each router is instructed to delete part of the useful information, rewrite the problem, and inject irrelevant noise. The outputs of multiple noisy routers are then aggregated and passed to the worker. In both conditions, the worker is not allowed to access the original question directly and must answer only on the basis of router outputs. The worker is also constrained to return a single option letter as the final answer. In this way, the difference between the clean and noisy conditions is restricted to the transmission regime itself, rather than to changes in answering rules or evaluation criteria.

4.4 Evaluation Metrics

Let the clean-baseline accuracy be the reference performance and the noisy-condition accuracy be the comparison performance. Their difference is defined as

$$\text{gain} = \text{noisy accuracy} - \text{clean baseline}.$$

Whenever $\text{gain} > 0$, the noisy condition outperforms the clean baseline. To isolate only the above-baseline component, we further define

$$\text{positive excess} = \max(\text{gain}, 0).$$

A noisy setting is counted as a violation whenever $\text{gain} > 0$. For a given model, the number of noisy settings under which violations occur defines its violation breadth, which summarizes how broadly the model's benchmark score is affected by potential contamination-related cues.

In addition to score-level metrics, we examine question-level transition directions. If a question is answered incorrectly under the clean baseline but correctly under a noisy condition, it is counted as an improve transition (wrong→correct). If a question is answered correctly under the clean baseline but incorrectly under a noisy condition, it is counted as a degrade transition (correct→wrong). These transition metrics help distinguish whether above-baseline anomalies are more consistent with random fluctuation or with a directional process in which noisy aggregation systematically helps recover benchmark answers.

5 Results and Analysis

This section evaluates the three hypotheses developed above. The empirical logic proceeds in three steps. We first examine whether noisy conditions genuinely produce above-baseline anomalies relative to the clean baseline. We then study whether such anomalies are heterogeneous across models. Finally, we move to question-level transition patterns to assess whether the observed gains are better understood as random fluctuation or as a directional process in which noisy aggregation systematically improves outcomes.

5.1 Overall violations: do noisy conditions really exceed the baseline?

Figure 2 shows the overall deviation of model performance from the clean baseline under different noisy-router settings. At the aggregate level, above-baseline anomalies are not isolated outliers. As the number of noisy routers increases from 1 to 9, the number of models exceeding the clean baseline is 5/12, 4/12, 6/12, 7/12, 7/12, 7/12, 8/12, 10/12, and 8/12, respectively. The highest violation count occurs at $r = 8$, where 10 out of 12 models rise above the baseline. The mean positive excess also becomes more pronounced in higher-router regions, reaching 0.066 at $r = 8$ and 0.086 at $r = 9$, the largest value across all settings.The complete router-level summary statistics are reported in Appendix Table 3.

These results support Hypothesis 1. If the clean baseline corresponds to the regime closest to full-information transmission, then noisy conditions should at most fluctuate around that baseline rather than persistently exceed it. The key pattern in Figure 2 is therefore not that "more routers make models stronger," but that higher-router settings make above-baseline anomalies both more frequent and more substantial. This is consistent with the theoretical argument that deleted, rewritten, and perturbed fragments may be recombined into semantic-neighbor cues that reactivate benchmark-related memory traces.

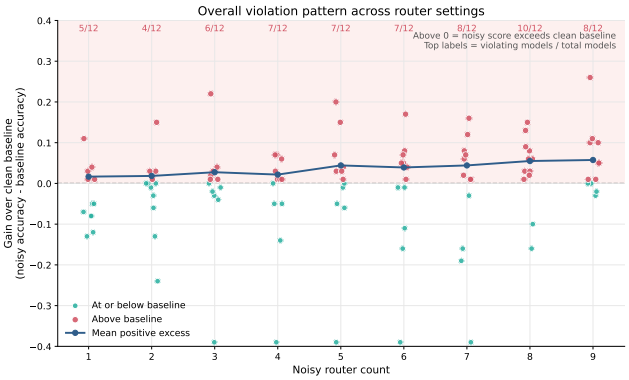

Figure 2: Overall violation pattern across router settings.

5.2 Model heterogeneity: models differ in the probability and magnitude of anomalous gains

Figure 3 plots model-specific performance trajectories relative to the clean baseline under different noisy-router settings. The anomalous-gain pattern is clearly heterogeneous across models rather than uniformly distributed. Some models exceed the baseline under almost all noisy settings. For example, Qwen3-Next-80B violates the baseline in all 9 router settings, while Seed-2.0-Lite does so in 8 out of 9. By contrast, DeepSeek-Chat exceeds the baseline only once, and Qwen3.5-122B does so only twice.

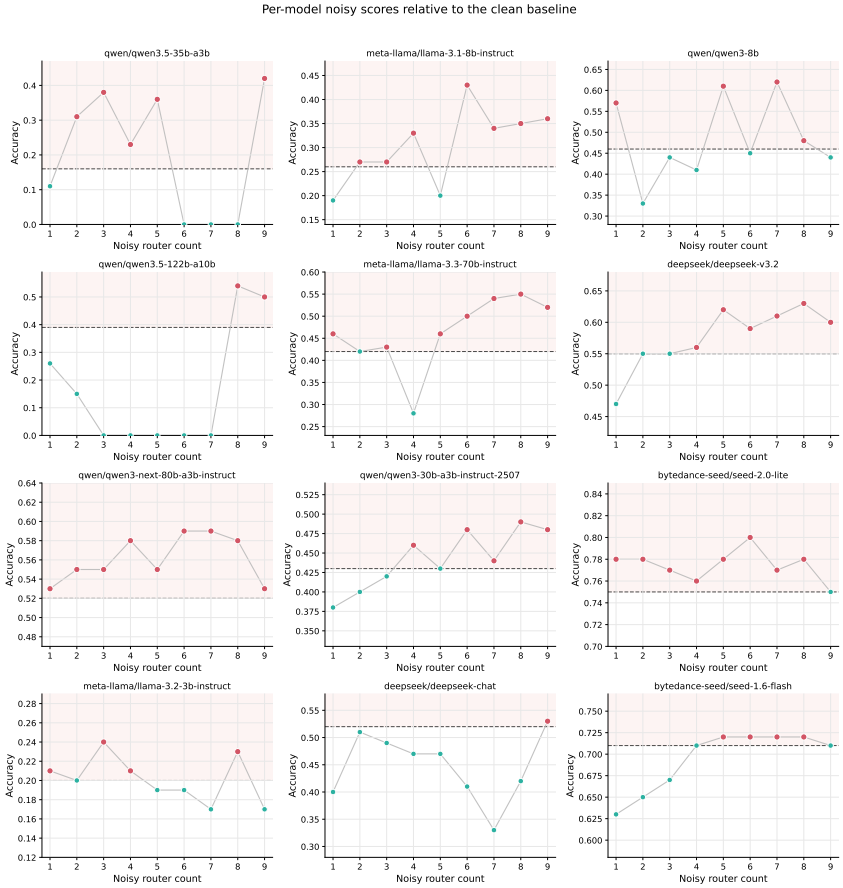

Figure 3: Per-model noisy scores relative to the clean baseline.

The breadth and strength of anomalies are also not identical. Llama-3.1-8B and Llama-3.3-70B both violate the baseline in 7 out of 9 settings, indicating relatively broad exposure. By contrast, Qwen3.5-35B violates the baseline in only 5 settings but reaches a maximum positive excess of 0.260, the highest among all models. Qwen3-Next-80B displays a different pattern: violations occur in all 9 settings, but the maximum jump is smaller. This suggests that contamination sensitivity has at least two dimensions: how often a model crosses the baseline, and how far above the baseline it moves once it does so.

These results support Hypothesis 2. Even when models obtain similar benchmark scores, their sensitivity to contamination-related cues may differ substantially. Model comparison should therefore not stop at score levels alone, but should also consider whether those scores are equally credible. For space reasons, the compressed model-level summary is deferred to the appendix: Appendix Figure 5 visualizes violation breadth across models, Appendix Table 1 reports the corresponding model-level summary statistics, and Appendix Table 2 provides the full model-by-router breakdown.

5.3 Question-level mechanism: random fluctuation or directional improvement?

Figure 4 examines transition directions at the question level. Here, improve denotes a wrong→correct transition, that is, a question answered incorrectly under the clean baseline but correctly under a noisy condition; degrade denotes a correct→wrong transition. The figure aggregates these transitions across all models, so the vertical axis represents the total number of question-level transitions rather than the number of questions for any single model.The figure aggregates these transitions across all models, so the vertical axis represents the total number of question-level transitions rather than the number of questions for any single model.

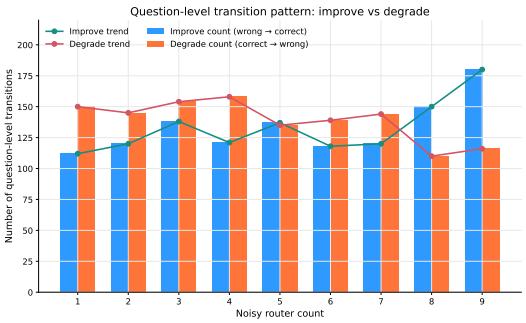

Figure 4: Question-level transition pattern: improve vs. degrade.

The improve and degrade curves are not strictly monotonic at every router count, but their overall movement is clearly directional. As the number of noisy routers increases from 1 to 9, improve rises from 112 to 180, whereas degrade falls from 150 to 116. More importantly, their relative ordering reverses in higher-router regions. At $r = 1$, improve is 38 cases lower than degrade; by $r = 5$, the two are nearly balanced (137 versus 135); and by $r = 8$ and $r = 9$, improve reaches 150 and 180, clearly exceeding degrade at 110 and 116. In other words, as noisy routers accumulate, the question-level pattern shifts from "more degraded than improved" to "more improved than degraded."

This evidence is consistent with Hypothesis 3. The claim is not that every additional router mechanically raises improve and lowers degrade at each single point, but that the overall trend increasingly favors wrong→correct transitions as router count grows, especially in higher-router conditions. This directional reversal suggests that noisy aggregation does not merely destroy information. It can also reconstruct semantically related cues that help recover benchmark answers, thereby turning previously incorrect responses into correct ones. The above-baseline anomaly is therefore reflected not only in score-level deviations but also in a question-level process of directional improvement.

6 Conclusion and Discussion

This study develops an audit framework for score confidence in LLM benchmarks. The results show that, at the aggregate level, above-baseline anomalies under noisy conditions are not isolated cases but recur across multiple router settings. At the model level, the breadth of violations and the magnitude of positive excess differ substantially across models, indicating that benchmark scores vary in their sensitivity to potential contamination-related cues. At the question level, as the number of noisy routers increases, wrong-to-correct transitions tend to rise, correct-to-wrong transitions tend to fall, and the former eventually exceeds the latter in higher-router conditions. Taken together, these findings suggest that benchmark scores are not merely simple records of answer correctness; they may also reflect a model's responsiveness to benchmark-related semantic-neighbor information. A

high score therefore does not necessarily imply a high-confidence score, and even models with similar benchmark performance may differ substantially in how credibly their scores represent genuine generalization ability.

These findings do not imply that benchmarks are entirely invalid. For LLMs, the ability to retrieve the correct answer within an institutionalized answering structure is itself a real and practically relevant form of competence. The problem is that such institutional success is often directly interpreted as evidence of genuine generalization ability. The core issue of Silicon Bureaucracy and AI Test-Oriented Education is therefore not the existence of examinations themselves, but the tendency to treat examination outcomes as pure ability. The implication is accordingly not to reject benchmarks, but to reinterpret them: what benchmarks measure needs to be reconsidered, and how benchmark scores are used requires additional auditing.

References

Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan AlRashed, Shaykhah Alsubaie, Yousef Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13787–13805, Bangkok, Thailand, 2024. Association for Computational Linguistics.

Hongjun An, Wenhan Hu, Sida Huang, Siqi Huang, Ruanjun Li, Yuanzhi Liang, Jiawei Shao, Yiliang Song, Zihan Wang, Cheng Yuan, Chi Zhang, Hongyuan Zhang, Wenhao Zhuang, and Xuelong Li. AI Flow: Perspectives, Scenarios, and Approaches. *Vicinagearth*, 3:1, 2026a.

Hongjun An, Yiliang Song, Jiangan Chen, Jiawei Shao, Chi Zhang, and Xuelong Li. Are LLMs Vulnerable to Preference-Undermining Attacks (PUA)? A Factorial Analysis Methodology for Diagnosing the Trade-off between Preference Alignment and Real-World Validity, 2026b.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. MT-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7421–7454, Bangkok, Thailand, 2024. Association for Computational Linguistics.

Andrew M. Bean, Ryan Othniel Kearns, Angelika Romanou, Franziska Sofia Hafner, Harry Mayne, Jan Batzner, Negar Foroutan, Chris Schmitz, Karolina Korgul, Hunar Batra, Oishi Deb, Emma Beharry, Cornelius Emde, Thomas Foster, Anna Gausen, María Grandury, Simeng Han, Valentin Hofmann, Lujain Ibrahim, Hazel Kim, Hannah Rose Kirk, Fangru Lin, Gabrielle Kaili-May Liu, Lennart Luettgau, Jabez Magomere, Jonathan Rystrøm, Anna Sotnikova, Yushi Yang, Yilun Zhao, Adel Bibi, Antoine Bosselut, Ronald Clark, Arman Cohan, Jakob Foerster, Yarin Gal, Scott A. Hale, Inioluwa Deborah Raji, Christopher Summerfield, Philip H. S. Torr, Cozmin Ududec, Luc Rocher, and Adam Mahdi. Measuring what matters: Construct validity in large language model benchmarks, 2025.

Dylan Bouchard, Mohit Singh Chauhan, David Skarbrevik, Ho-Kyeong Ra, Viren Bajaj, and Zeya Ahmad. UQLM: A python package for uncertainty quantification in large language models. *Journal of Machine Learning Research*, 27(13):1–10, 2026.

Center for AI Safety, Scale AI, and HLE Contributors Consortium. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649:1139–1146, 2026.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38 (16):17754–17762, 2024.

Jasper Dekoninck, Mark Niklas Müller, and Martin Vechev. Constat: Performance-based contamination detection in large language models. In *Advances in Neural Information Processing Systems 37*, 2024.

Shulin Huang, Linyi Yang, Yan Song, Shuang Chen, Leyang Cui, Ziyu Wan, Qingcheng Zeng, Ying Wen, Kun Shao, Weinan Zhang, Jun Wang, and Yue Zhang. Thinkbench: Dynamic out-of-distribution evaluation for robust LLM reasoning, 2025a.

Zhongzhan Huang, Guoming Ling, Shanshan Zhong, Hefeng Wu, and Liang Lin. Minilongbench: The low-cost long context understanding benchmark for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11442–11460, Vienna, Austria, 2025b. Association for Computational Linguistics.

Shreya Johri, Jaehwan Jeong, Benjamin A. Tran, Daniel I. Schlessinger, Shannon Wongvibulsin, Leandra A. Barnes, Hong-Yu Zhou, Zhuo Ran Cai, Eliezer M. Van Allen, David Kim, Roxana Daneshjou, and Pranav Rajpurkar. An evaluation framework for clinical use of large language models in patient interaction tasks. *Nature Medicine*, 31:77–86, 2025.

Wai-Chung Kwan, Xingshan Zeng, Yufei Wang, Yusen Sun, Liangyou Li, Yuxin Jiang, Lifeng Shang, Qun Liu, and Kam-Fai Wong. M4LE: A multi-ability multi-range multi-task multi-domain long-context evaluation benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15568–15592, Bangkok, Thailand, 2024. Association for Computational Linguistics.

Xiang Li, Yunshi Lan, and Chao Yang. Treeeval: Benchmark-free evaluation of large language models through tree planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (23):24485–24493, 2025.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems 36*, 2023.

Shiwen Ni, Xiangtao Kong, Chengming Li, Xiping Hu, Ruifeng Xu, Jia Zhu, and Min Yang. Training on the benchmark is not all you need. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24948–24956, 2025.

Thao Pham. Truth behind the scene: Designing evaluations benchmarks to assess LLMs' task-specific understanding over test-taking strategies. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28):29596–29598, 2025.

Jiawei Shao and Xuelong Li. AI Flow at the Network Edge. *IEEE Network*, 2025.

Anna Sokol, Elizabeth Daly, Michael Hind, David Piorkowski, Xiangliang Zhang, Nuno Moniz, and Nitesh V. Chawla. Benchmarkcards: Standardized documentation for large language model benchmarks, 2024.

Huan Song, Qingfei Zhao, Ting Long, Shuyu Tian, Hongjun An, Jiawei Shao, and Xuelong Li. Theoretical foundations of scaling law in familial models. *arXiv preprint arXiv:2512.23407*, 2025.

Huan Song, Shuyu Tian, Junyi Hao, Minxiu Xu, Hongjun An, Yiliang Song, Jiawei Shao, and Xuelong Li. Ruyi2 Technical Report. *arXiv preprint arXiv:2602.22543*, 2026a.

Yiliang Song, Hongjun An, Jiangong Xiao, Haofei Zhao, Jiawei Shao, and Xuelong Li. Creditaudit: 2d auditing for LLM evaluation and selection, 2026b.

Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. SciEval: A multi-level large language model evaluation benchmark for scientific research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19053–19061, 2024.

Yifan Sun, Han Wang, Dongbai Li, Gang Wang, and Huan Zhang. The emperor's new clothes in benchmarking? a rigorous examination of mitigation strategies for LLM benchmark data contamination. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 57728–57753. PMLR, 2025.

Zhuo Wang, Wen Wu, Guoqing Wang, Guangze Ye, and Zhenxiao Cheng. Metaeval: Measuring the discrimination of benchmarks for efficient LLM evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(40):33773–33781, 2026.

Junjie Ye, Zhengyin Du, Xuesong Yao, Weijian Lin, Yufei Xu, Zehui Chen, Zaiyuan Wang, Sining Zhu, Zhiheng Xi, Siyu Yuan, Tao Gui, Qi Zhang, Xuanjing Huang, and Jiecao Chen. Toolhop: A query-driven benchmark for evaluating large language models in multi-hop tool use. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2995–3021, Vienna, Austria, 2025. Association for Computational Linguistics.

Cheng Yuan, Jiawei Shao, and Xuelong Li. Information Capacity: Evaluating the Efficiency of Large Language Models via Text Compression. *arXiv preprint arXiv:2511.08066*, 2025.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue. A careful examination of large language model performance on grade school arithmetic. In *Advances in Neural Information Processing Systems 37*, 2024a.

Zhehao Zhang, Jiaao Chen, and Diyi Yang. DARG: Dynamic evaluation of large language models via adaptive reasoning graph evolvement. In *Advances in Neural Information Processing Systems 37*, 2024b.

Yilun Zhao, Weiyuan Chen, Zhijian Xu, Manasi Patwardhan, Chengye Wang, Yixin Liu, Lovekesh Vig, and Arman Cohan. Abgen: Evaluating large language models in ablation study design and evaluation for scientific research. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12479–12491, Vienna, Austria, 2025. Association for Computational Linguistics.

Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. Promptbench: A unified library for evaluation of large language models. *Journal of Machine Learning Research*, 25(254):1–22, 2024.

A Technical appendices and supplementary material

This appendix provides supplementary evidence that complements the main text at both the model level and the router level. It includes one additional figure and three additional tables. Together, these materials provide the full numerical background for the heterogeneity patterns discussed in Section 5.2 and the router-level and question-level transition results discussed in Sections 5.1 and 5.3.

Table 1 reports model-level summary statistics, including violation breadth and positive-excess measures.

Table 1: Model-level summary of violation breadth and positive excess

Model	Viol. Count	Viol. Rate	Max Pos. Excess	Mean Pos. Excess	Mean Gain
Qwen3-Next-80B	9/9	1.000	0.070	0.041	0.041
Seed-2.0-Lite	8/9	0.889	0.050	0.028	0.024
Llama-3.1-8B	7/9	0.778	0.170	0.076	0.044
Llama-3.3-70B	7/9	0.778	0.130	0.074	0.042
DeepSeek-V3.2	6/9	0.667	0.080	0.052	0.026
Qwen3.5-35B	5/9	0.556	0.260	0.178	0.040
Qwen3-30B	5/9	0.556	0.060	0.040	0.012
Qwen3-8B	4/9	0.444	0.160	0.110	0.023
Llama-3.2-3B	4/9	0.444	0.040	0.022	0.001
Seed-1.6-Flash	4/9	0.444	0.010	0.010	-0.016
Qwen3.5-122B	2/9	0.222	0.150	0.130	-0.229
DeepSeek-Chat	1/9	0.111	0.010	0.010	-0.072

Table 2 reports the full model-by-router breakdown underlying the anomalous-gain patterns discussed in the main text.

Table 2: Model-by-router violation details

Model	Router	Clean	Noisy	Gain	Improve	Degrade
Seed-1.6-Flash	1	0.71	0.63	-0.08	8	16
Seed-1.6-Flash	2	0.71	0.65	-0.06	6	12
Seed-1.6-Flash	3	0.71	0.67	-0.04	8	12
Seed-1.6-Flash	4	0.71	0.71	0.00	6	6
Seed-1.6-Flash	5	0.71	0.72	0.01	7	6
Seed-1.6-Flash	6	0.71	0.72	0.01	6	5
Seed-1.6-Flash	7	0.71	0.72	0.01	6	5
Seed-1.6-Flash	8	0.71	0.72	0.01	5	4
Seed-1.6-Flash	9	0.71	0.71	0.00	6	6
Seed-2.0-Lite	1	0.75	0.78	0.03	6	3
Seed-2.0-Lite	2	0.75	0.78	0.03	6	3
Seed-2.0-Lite	3	0.75	0.77	0.02	5	3
Seed-2.0-Lite	4	0.75	0.76	0.01	5	4
Seed-2.0-Lite	5	0.75	0.78	0.03	4	1
Seed-2.0-Lite	6	0.75	0.80	0.05	5	0
Seed-2.0-Lite	7	0.75	0.77	0.02	4	2
Seed-2.0-Lite	8	0.75	0.78	0.03	5	2
Seed-2.0-Lite	9	0.75	0.75	0.00	5	5
DeepSeek-Chat	1	0.52	0.40	-0.12	5	17
DeepSeek-Chat	2	0.52	0.51	-0.01	10	11
DeepSeek-Chat	3	0.52	0.49	-0.03	13	16
DeepSeek-Chat	4	0.52	0.47	-0.05	10	15
DeepSeek-Chat	5	0.52	0.47	-0.05	9	14
DeepSeek-Chat	6	0.52	0.41	-0.11	9	20
DeepSeek-Chat	7	0.52	0.33	-0.19	6	25
DeepSeek-Chat	8	0.52	0.42	-0.10	7	17
DeepSeek-Chat	9	0.52	0.53	0.01	11	10
DeepSeek-V3.2	1	0.55	0.47	-0.08	7	15
DeepSeek-V3.2	2	0.55	0.55	0.00	9	9
DeepSeek-V3.2	3	0.55	0.55	0.00	11	11
DeepSeek-V3.2	4	0.55	0.56	0.01	10	9
DeepSeek-V3.2	5	0.55	0.62	0.07	17	10
DeepSeek-V3.2	6	0.55	0.59	0.04	14	10
DeepSeek-V3.2	7	0.55	0.61	0.06	13	7
DeepSeek-V3.2	8	0.55	0.63	0.08	16	8
DeepSeek-V3.2	9	0.55	0.60	0.05	13	8
Llama-3.1-8B	1	0.26	0.19	-0.07	6	13
Llama-3.1-8B	2	0.26	0.27	0.01	11	10
Llama-3.1-8B	3	0.26	0.27	0.01	17	16
Llama-3.1-8B	4	0.26	0.33	0.07	17	10
Llama-3.1-8B	5	0.26	0.20	-0.06	11	17
Llama-3.1-8B	6	0.26	0.43	0.17	21	4
Llama-3.1-8B	7	0.26	0.34	0.08	18	10
Llama-3.1-8B	8	0.26	0.35	0.09	18	9
Llama-3.1-8B	9	0.26	0.36	0.10	17	7
Llama-3.2-3B	1	0.20	0.21	0.01	7	6
Llama-3.2-3B	2	0.20	0.20	0.00	5	5
Llama-3.2-3B	3	0.20	0.24	0.04	10	6
Llama-3.2-3B	4	0.20	0.21	0.01	9	8
Llama-3.2-3B	5	0.20	0.19	-0.01	4	5
Llama-3.2-3B	6	0.20	0.19	-0.01	7	8
Llama-3.2-3B	7	0.20	0.17	-0.03	9	12
Llama-3.2-3B	8	0.20	0.23	0.03	8	5
Llama-3.2-3B	9	0.20	0.17	-0.03	8	11
Llama-3.3-70B	1	0.42	0.46	0.04	10	6
Llama-3.3-70B	2	0.42	0.42	0.00	11	11
Llama-3.3-70B	3	0.42	0.43	0.01	15	14
Llama-3.3-70B	4	0.42	0.28	-0.14	5	19

Continued on next page

Table 2 continued from previous page

Model	Router	Clean	Noisy	Gain	Improve	Degrade
Llama-3.3-70B	5	0.42	0.46	0.04	15	11
Llama-3.3-70B	6	0.42	0.50	0.08	15	7
Llama-3.3-70B	7	0.42	0.54	0.12	17	5
Llama-3.3-70B	8	0.42	0.55	0.13	17	4
Llama-3.3-70B	9	0.42	0.52	0.10	17	7
Qwen3-30B	1	0.43	0.38	-0.05	7	12
Qwen3-30B	2	0.43	0.40	-0.03	9	12
Qwen3-30B	3	0.43	0.42	-0.01	9	10
Qwen3-30B	4	0.43	0.46	0.03	13	10
Qwen3-30B	5	0.43	0.43	0.00	7	7
Qwen3-30B	6	0.43	0.48	0.05	13	8
Qwen3-30B	7	0.43	0.44	0.01	10	9
Qwen3-30B	8	0.43	0.49	0.06	14	8
Qwen3-30B	9	0.43	0.48	0.05	15	10
Qwen3-8B	1	0.46	0.57	0.11	23	12
Qwen3-8B	2	0.46	0.33	-0.13	12	25
Qwen3-8B	3	0.46	0.44	-0.02	15	17
Qwen3-8B	4	0.46	0.41	-0.05	17	22
Qwen3-8B	5	0.46	0.61	0.15	22	7
Qwen3-8B	6	0.46	0.45	-0.01	16	17
Qwen3-8B	7	0.46	0.62	0.16	24	8
Qwen3-8B	8	0.46	0.48	0.02	20	18
Qwen3-8B	9	0.46	0.44	-0.02	17	19
Qwen3-Next-80B	1	0.52	0.53	0.01	8	7
Qwen3-Next-80B	2	0.52	0.55	0.03	8	5
Qwen3-Next-80B	3	0.52	0.55	0.03	7	4
Qwen3-Next-80B	4	0.52	0.58	0.06	11	5
Qwen3-Next-80B	5	0.52	0.55	0.03	11	8
Qwen3-Next-80B	6	0.52	0.59	0.07	12	5
Qwen3-Next-80B	7	0.52	0.59	0.07	13	6
Qwen3-Next-80B	8	0.52	0.58	0.06	11	5
Qwen3-Next-80B	9	0.52	0.53	0.01	7	6
Qwen3.5-122B	1	0.39	0.26	-0.13	15	28
Qwen3.5-122B	2	0.39	0.15	-0.24	8	32
Qwen3.5-122B	3	0.39	0.00	-0.39	0	39
Qwen3.5-122B	4	0.39	0.00	-0.39	0	39
Qwen3.5-122B	5	0.39	0.00	-0.39	0	39
Qwen3.5-122B	6	0.39	0.00	-0.39	0	39
Qwen3.5-122B	7	0.39	0.00	-0.39	0	39
Qwen3.5-122B	8	0.39	0.54	0.15	29	14
Qwen3.5-122B	9	0.39	0.50	0.11	26	15
Qwen3.5-35B	1	0.16	0.11	-0.05	10	15
Qwen3.5-35B	2	0.16	0.31	0.15	25	10
Qwen3.5-35B	3	0.16	0.38	0.22	28	6
Qwen3.5-35B	4	0.16	0.23	0.07	18	11
Qwen3.5-35B	5	0.16	0.36	0.20	30	10
Qwen3.5-35B	6	0.16	0.00	-0.16	0	16
Qwen3.5-35B	7	0.16	0.00	-0.16	0	16
Qwen3.5-35B	8	0.16	0.00	-0.16	0	16
Qwen3.5-35B	9	0.16	0.42	0.26	38	12

Table 3 reports router-level violation statistics together with the corresponding question-level transition counts.

Table 3: Router-level violation statistics and transition counts

Router	Viol. Models	Viol. Rate	Mean Pos. Excess	Improve	Degrade	Net Improve
1	5/12	0.417	0.040	112	150	-38
2	4/12	0.333	0.055	120	145	-25
3	6/12	0.500	0.055	138	154	-16
4	7/12	0.583	0.037	121	158	-37
5	7/12	0.583	0.076	137	135	2
6	7/12	0.583	0.067	118	139	-21
7	8/12	0.667	0.066	120	144	-24
8	10/12	0.833	0.066	150	110	40
9	8/12	0.667	0.086	180	116	64

Figure 5 provides a compact model-level visualization of violation breadth.

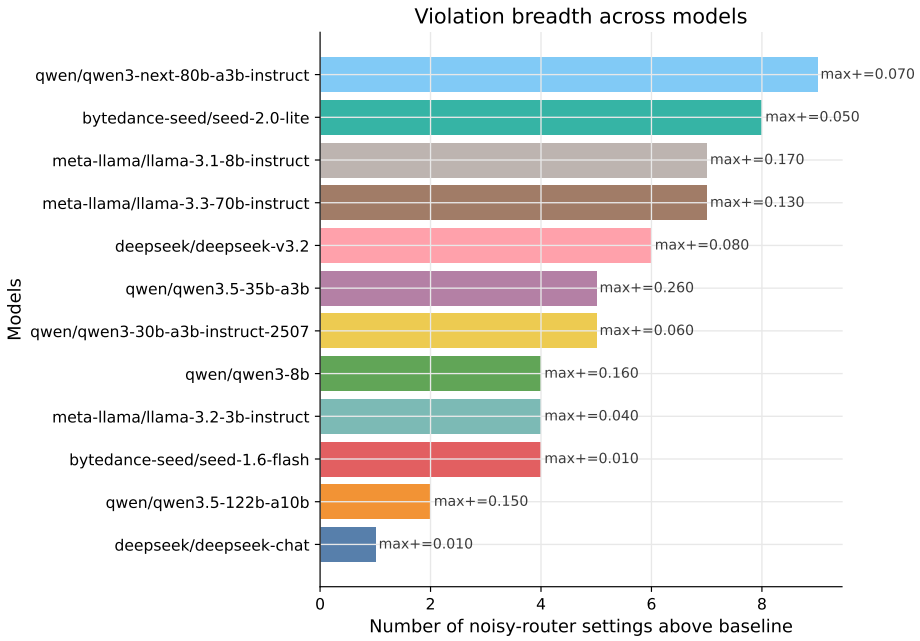

Figure 5: Violation breadth across models.