Static Benchmarks Are Broken: The Case for Dynamic Evaluation of LLMs

Farhan Ahmed [1] **Chad DeLuca** [1]

Abstract

Static benchmarks have become the primary tool for measuring large language model (LLM) progress, yet growing evidence suggests they increasingly reward memorization and surface-form pattern-matching over genuine capability. Performance on canonical benchmarks such as MMLU and GSM8k degrades sharply under semantics-preserving perturbations, including answer reordering, surface rephrasing, and distractor addition, a brittleness inconsistent with the robust understanding these benchmarks are meant to certify. We argue this fragility is not an implementation flaw but a structural consequence of fixed evaluation sets in the era of web-scale training. We advocate for dynamic, synthetically generated benchmarks constructed fresh at evaluation time, eliminating instance-level contamination by construction and enabling principled, reproducible evaluation of genuine model capability. The remaining weaknesses of dynamic evaluation are tractable; those of static evaluation are structural.

1. Introduction

Benchmarks have become the de facto measure of progress in large language model (LLM) research (Hendrycks et al., 2021; Cobbe et al., 2021). Leaderboard rankings drive research directions, inform product decisions, and serve as proxies for real-world capability (Fourrier et al., 2023; 2024). This reliance rests on a critical assumption: that high performance on a static benchmark reflects the underlying skill it is designed to measure. The cost of this assumption failing is not abstract: when reported gains reflect contamination rather than capability, the field optimizes toward leaderboard position while real-world reliability stagnates, and benchmark rankings increasingly diverge from how models are judged in deployment (Chiang et al., 2024).

[1]IBM Research. Correspondence to: Farhan Ahmed <farhan.ahmed@ibm.com>.

Accepted at ICML 2026 Workshop on Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning.

We argue this assumption is broken. As LLMs are trained on increasingly vast web-scale datasets, the boundary between training and evaluation data erodes (Mirzadeh et al., 2025; Wang et al., 2024a). Static benchmarks, which consist of fixed question sets with predetermined answers, become inadvertent training targets. Even without deliberate data contamination, models tuned to maximize performance on well-known benchmarks such as MMLU (Hendrycks et al., 2021) and GSM8k (Cobbe et al., 2021) learn to pattern-match against memorized answer forms rather than develop genuine understanding (Zheng et al., 2024; Pezeshkpour & Hruschka, 2023).

The implication is not that these benchmarks need more careful curation, but that the static paradigm itself has reached its limit. This paper presents evidence that this failure mode is already observable, argues that structural features of static benchmarks make it unavoidable, and advocates for a shift toward dynamic, synthetically generated evaluation as the only principled remedy. The pathology is not unique to language understanding: offline decision-making settings, from reinforcement learning to black-box optimization, face the same hazard whenever policies or optimizers are selected against a fixed evaluation set, and the same parameterized, seeded generation we advocate is the natural remedy there.

2. The Fragility of Static Benchmarks

Multiple-Choice Benchmarks MMLU (Hendrycks et al., 2021) has become the canonical benchmark for measuring LLM knowledge and reasoning across 57 tasks, yet its scores are highly sensitive to evaluation artifacts unrelated to the knowledge being tested. Permuting the order of answer choices causes substantial accuracy drops, with some models losing over 10 percentage points (Pezeshkpour & Hruschka, 2023; Zheng et al., 2024), an effect confirmed at scale (Gupta et al., 2024). Rephrasing questions or introducing linguistic variation causes further degradation even when factual content is unchanged (Wang et al., 2024a). The multiple-choice format itself conflates test-taking heuristics with the target capability, making it a poor proxy for genuine understanding (Li et al., 2024b). A model with genuine knowledge should be invariant to distractor ordering; such sensitivity is the empirical signature of positional bias and memorization, not understanding.

Mathematical Reasoning Benchmarks GSM8k (Cobbe et al., 2021) is the standard benchmark for grade-school mathematical reasoning, widely used to evaluate chain-of-thought arithmetic capabilities (Wei et al., 2022). GSM-Symbolic (Mirzadeh et al., 2025) replaces names and numerical values in GSM8k problems while preserving the underlying mathematical structure. Performance drops significantly across all tested models, with variance increasing as problem complexity grows. A broader robustness evaluation through GSM-Plus confirms systematic degradation across diverse perturbation types, including numerical substitution, question rephrasing, and the addition of irrelevant clauses (Li et al., 2024a). The mathematical structure is identical across original and perturbed variants; only the surface form changes. A model reasoning from first principles should solve both equally well. The performance gap reveals that models are recovering memorized solution templates keyed to surface features, not solving the abstract problem (Mirzadeh et al., 2025; Li et al., 2024a). Both benchmark types exhibit the same failure mode: high performance on the canonical evaluation form, brittleness to semantics-preserving variation, invariance to meaning and sensitivity to form. That this pattern appears independently across benchmark types, research groups, and model families (Pezeshkpour & Hruschka, 2023; Zheng et al., 2024; Gupta et al., 2024; Wang et al., 2024a; Mirzadeh et al., 2025; Li et al., 2024a) suggests it reflects a systematic property of static evaluation rather than isolated benchmark flaws.

3. Static Benchmarks Cannot Fix Themselves

The structural argument applies to any fixed evaluation set regardless of domain or difficulty; the same fragility has been documented in reading comprehension (Jia & Liang, 2017) and leaderboard saturation across benchmark types confirms it is pervasive (Fourrier et al., 2023; 2024) (see Appendix A). The natural response to these findings is to construct harder static benchmarks with stricter contamination controls. MMLU-Pro (Wang et al., 2024b) attempts exactly this by expanding the choice set to ten options and curating harder problems. But this approach treats the symptom rather than the cause.

The fundamental problem is structural: a static benchmark is a closed dataset. Once its contents enter training pipelines, even indirectly through web crawls, distillation, or synthetic data generation, no amount of curation restores its validity as an unbiased measure (Jacovi et al., 2023; Mirzadeh et al., 2025; Wang et al., 2024a). Benchmark updates simply reset the clock; the replacement eventually becomes contaminated in turn, a cycle already evident in the rapid saturation of successive MMLU variants (Fourrier et al., 2023; 2024).

Furthermore, fixed answer sets directly incentivize memorization during fine-tuning. Greedy decoding against a predetermined correct answer is, mechanistically, next-token prediction on memorized text (Mirzadeh et al., 2025; Zheng et al., 2024). This is not a solvable problem within the static benchmark paradigm.

4. Dynamic Benchmarks as the Path Forward

We advocate for *dynamic benchmarks*: evaluation instances generated fresh at evaluation time from a parameterized generative process, rather than drawn from a fixed held-out set. Dynamic generation is contamination-resistant by construction, since it is not possible to memorize an instance that does not exist until the moment of evaluation (Jacovi et al., 2023; Castillo-Bolado et al., 2024). Two existing benchmarks demonstrate that this is feasible at scale. RULER (Hsieh et al., 2024) evaluates long-context capabilities through synthetically generated needle-in-a-haystack and retrieval tasks, with ground truth derived algorithmically from the generated context. LongReason (Ling et al., 2025) constructs synthetic multi-step reasoning problems through context expansion. Both benchmarks produce meaningful, fine-grained capability signals, providing evidence that synthetic generation does not sacrifice evaluation quality (Hsieh et al., 2024; Ling et al., 2025). The paradigm is not specific to language: procedurally generated reinforcement learning benchmarks (Cobbe et al., 2020) and parameterized black-box optimization suites (Hansen et al., 2020) have long evaluated agents and optimizers on freshly sampled instances rather than a fixed set, indicating that seeded dynamic evaluation is a general methodological commitment rather than a domain-specific trick.

Beyond contamination resistance, dynamic benchmarks offer structural properties that static benchmarks typically cannot provide:

- **Tunable difficulty.** Generative parameters directly control problem complexity, enabling controlled evaluation across the capability spectrum (Hsieh et al., 2024; Ling et al., 2025).
- **Verifiable ground truth.** Domains with algorithmic answers, symbolic reasoning, code execution, and formal logic, admit objective correctness without human annotation (Hsieh et al., 2024; Maheshwari et al., 2024).
- **Reproducibility via seeding.** Dynamic does not mean irreproducible. Versioned random seeds produce identical instance distributions for all evaluators, preserving comparability without exposing instances in advance (Hsieh et al., 2024; Castillo-Bolado et al., 2024).

The reproducibility objection to dynamic benchmarks is therefore an implementation concern, not a principled one. Complementary evidence shows that dynamic and synthetic evaluation frameworks are both practically viable and empirically informative across diverse task settings (Castillo-

Table 1. Structural comparison of static and dynamic benchmark paradigms across evaluation-relevant properties.

Property	Static Benchmarks	Dynamic Benchmarks
Contamination risk	High; fixed instances can enter training corpora through web crawls, distillation, or synthetic augmentation (Jacovi et al., 2023)	Instance-level: none by construction, as instances do not exist prior to evaluation time (generator-level overfitting remains a design concern)
Reproducibility	Exact; all evaluators use identical instances	Exact with versioned random seeds; results require seed and generator version to be published (Hsieh et al., 2024)
Difficulty control	Fixed at dataset creation; no post-hoc adjustment	Tunable via generative parameters; enables controlled evaluation across capability levels (Hsieh et al., 2024; Ling et al., 2025)
Ground truth	Human-curated; subject to annotation noise and subjectivity	Algorithmic; derived directly from generation parameters for suitable domains (Maheshwari et al., 2024)
Long-term validity	Degrades as models saturate the fixed instance set (Fourrier et al., 2023; 2024)	Maintained; fresh instances generated at each evaluation
Task coverage	Broad, including open-ended and subjective tasks	Currently limited to domains with verifiable ground truth; open-ended tasks require human-in-the-loop (Zheng et al., 2023)
Evaluation cost	Low; fixed corpus evaluated once and reused	Higher; generation overhead at each evaluation run
Cross-study comparability	High when using the same dataset version	Requires standardized generators and versioning; variance from generator choice is a known risk (Liang et al., 2022)

Bolado et al., 2024; Aluffi et al., 2025; Maheshwari et al., 2024). Table 1 contrasts the two paradigms across the evaluation-relevant properties discussed here, together with the limitations detailed in Section 5.

Dynamic Evaluation as Statistical Estimation Framing generation statistically clarifies why contamination is not merely mitigated but structurally dissolved. A generator with a seed distribution induces a distribution $\mathcal{D}$ over task instances, and drawing N fresh instances yields an *unbiased* Monte Carlo estimate $\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{correct}(f, x_i)$ of a model f's true expected capability $\mu = \mathbb{E}_{x\sim\mathcal{D}}[\mathrm{correct}(f, x)]$. Because the instances are i.i.d. and, by construction, never observed during training, standard concentration applies: $|\hat{\mu} - \mu| \leq \sqrt{\ln(2/\delta)/(2N)}$ with probability at least $1 - \delta$, a genuine finite-sample confidence interval on capability. A fixed test set admits no analogous guarantee. Contamination makes its empirical accuracy a biased estimate of μ with bias of unknown sign and magnitude, and reusing it across rounds of model selection violates the independence its validity rests on, the familiar mechanism by which benchmark scores drift above true capability under repeated reuse (Recht et al., 2019). Dynamic generation also exposes the *capability surface* $\mu(\theta)$ as a function of the generator's difficulty parameters θ, turning evaluation from a single point estimate into a distribution over a controlled task family, precisely the object required for PAC-style statements over that family and for the offline-to-online generalization questions that arise whenever a system is evaluated on freshly drawn instances. These guarantees are only as broad as the generator's support: the bounds certify capability *on* $\mathcal{D}$, so the coverage and entropy of the generative process determine what the certificate actually means, the statistical counterpart of the generator-level overfitting that any single generator still permits.

5. Limitations of Dynamic Benchmarks

Dynamic evaluation is not a universal solution, and the case for it is stronger for being explicit about where it is currently weak.

Coverage of open-ended tasks. Automatic generation and verification work best for domains with algorithmic ground truth, as demonstrated by RULER (Hsieh et al., 2024) and LongReason (Ling et al., 2025). Tasks requiring subjective judgment, nuanced argumentation, or creative output are harder to generate and harder to evaluate without human involvement; work on LLM-as-judge evaluation shows that automated scoring of open-ended outputs systematically diverges from human judgments (Zheng et al., 2023), making human-in-the-loop validation necessary in these settings.

Validity of synthetic instances. A generative procedure may produce instances that are syntactically valid but semantically degenerate, including ambiguous, trivially easy, or unrepresentative problems. The efficacy of synthetic data as a reliable benchmark signal is not guaranteed and requires careful empirical validation (Maheshwari et al., 2024; Aluffi et al., 2025); the absence of human curation eliminates contamination but introduces new failure modes around instance quality.

Comparability across generators. Seeded generation ensures reproducibility within a study, but comparing results across labs using different generators or generator versions remains non-trivial. Holistic evaluation frameworks have shown that evaluation decisions, including metric choice, prompt format, and dataset construction, produce large variance in reported scores across otherwise comparable systems (Liang et al., 2022); dynamic benchmarks face the same risk at the generator level.

6. Implications and Call to Action

The community's reliance on static benchmarks has produced a problematic feedback loop: models are optimized against known evaluation sets, reported scores rise, and the field interprets this as capability progress (Fourrier et al., 2023; 2024; Wang et al., 2024b). Breaking this loop requires changing what we measure, not merely how carefully we measure it. We call for three concrete shifts: benchmark creators should publish *generative procedures* alongside fixed datasets; evaluation frameworks should adopt versioned seeding as a reproducibility standard; and the community should prioritize domains with algorithmic ground truth for initial dynamic evaluation suites. Real-world deployment does not present models with questions drawn from a fixed academic dataset, and human preference rankings diverge substantially from static benchmark scores (Chiang et al., 2024). Evaluation should reflect that reality.

7. Conclusion

The fragility of static benchmarks and the promise of dynamic ones reflect a deeper asymmetry, which is the core of our position. The properties on which static benchmarks fall short, contamination risk, long-term validity, difficulty control, and ground-truth reliability, are consequences of fixing the evaluation set that no amount of curation or harder problem selection can resolve. The properties on which dynamic benchmarks are currently weaker, principally task coverage and evaluation cost, are engineering and scope challenges that recede as generative and verification methods mature (Castillo-Bolado et al., 2024; Maheshwari et al., 2024). The limitations of dynamic benchmarks are tractable; those of static benchmarks are structural. We do not argue for abandoning static benchmarks overnight, but for beginning where dynamic evaluation is already viable, those domains with algorithmic ground truth, and expanding outward as the tooling matures. The deeper shift the field must make is to treat genuine, contamination-resistant measurement of capability, rather than ever-higher scores on a fixed set, as the standard it holds itself to.

References

Aluffi, P. A., Zietkiewicz, P., Bazzi, M., Arderne, M., and Murevics, V. Dynamic benchmarking framework for llm-based conversational data capture, 2025. URL https://arxiv.org/abs/2502.04349.

Castillo-Bolado, D., Davidson, J., Gray, F., and Rosa, M. Beyond prompts: Dynamic conversational benchmarking of large language models, 2024. URL https://arxiv.org/abs/2409.20222.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J. E., and Stoica, I. Chatbot arena: An open platform for evaluating LLMs by human preference, 2024. URL https://arxiv.org/abs/2403.04132.

Cobbe, K., Hesse, C., Hilton, J., and Schulman, J. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Fourrier, C., Habib, N., Lozovskaya, A., Szafer, K., and Wolf, T. OpenLLM Leaderboard V1. https://huggingface.co/docs/leaderboards/en/open_llm_leaderboard/archive, 2023.

Fourrier, C., Habib, N., Lozovskaya, A., Szafer, K., and Wolf, T. OpenLLM Leaderboard V2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.

Gupta, V., Pantoja, D., Ross, C., Williams, A., and Ung, M. Changing answer order can decrease mmlu accuracy, 2024. URL https://arxiv.org/abs/2406.19470.

Hansen, N., Auger, A., Ros, R., Mersmann, O., Tušar, T., and Brockhoff, D. Coco: a platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1):114–144, August

2020. ISSN 1029-4937. doi: 10.1080/10556788.2020.1808977. URL http://dx.doi.org/10.1080/10556788.2020.1808977.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., Zhang, Y., and Ginsburg, B. Ruler: What's the real context size of your long-context language models?, 2024. URL https://arxiv.org/abs/2404.06654.

Jacovi, A., Caciularu, A., Goldman, O., and Goldberg, Y. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks, 2023. URL https://arxiv.org/abs/2305.10160.

Jia, R. and Liang, P. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017. URL https://arxiv.org/abs/1707.07328.

Li, Q., Cui, L., Zhao, X., Kong, L., and Bi, W. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers, 2024a. URL https://arxiv.org/abs/2402.19255.

Li, W., Li, L., Xiang, T., Liu, X., Deng, W., and Garcia, N. Can multiple-choice questions really be useful in detecting the abilities of llms?, 2024b. URL https://arxiv.org/abs/2403.17752.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Manning, C. D., Ré, C., Acosta-Navas, D., Hudson, D. A., Zelikman, E., Durmus, E., Ladhak, F., Rong, F., Ren, H., Yao, H., Wang, J., Santhanam, K., Orr, L., Zheng, L., Yuksekgonul, M., Suzgun, M., Kim, N., Guha, N., Chatterji, N., Khattab, O., Henderson, P., Huang, Q., Chi, R., Xie, S. M., Hashimoto, T., Wu, T., Zhang, T., Chaudhary, V., Wang, W., Li, X., Mai, Y., Zhang, Y., and Koreeda, Y. Holistic evaluation of language models, 2022. URL https://arxiv.org/abs/2211.09110.

Ling, Z., Liu, K., Yan, K., Yang, Y., Lin, W., Fan, T.-H., Shen, L., Du, Z., and Chen, J. Longreason: A synthetic long-context reasoning benchmark via context expansion, 2025. URL https://arxiv.org/abs/2501.15089.

Maheshwari, G., Ivanov, D., and Haddad, K. E. Efficacy of synthetic data as a benchmark, 2024. URL https://arxiv.org/abs/2409.11968.

Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., and Farajtabar, M. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models, 2025. URL https://arxiv.org/abs/2410.05229.

Pezeshkpour, P. and Hruschka, E. Large language models sensitivity to the order of options in multiple-choice questions, 2023. URL https://arxiv.org/abs/2308.11483.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do ImageNet classifiers generalize to ImageNet? In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5389–5400. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/recht19a.html.

Wang, W., Jain, S., Kantor, P., Feldman, J., Gallos, L., and Wang, H. Mmlu-sr: A benchmark for stress-testing reasoning capability of large language models, 2024a. URL https://arxiv.org/abs/2406.15468.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., Li, T., Ku, M., Wang, K., Zhuang, A., Fan, R., Yue, X., and Chen, W. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024b. URL https://arxiv.org/abs/2406.01574.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2201.11903.

Zheng, C., Zhou, H., Meng, F., Zhou, J., and Huang, M. Large language models are not robust multiple choice selectors, 2024. URL https://arxiv.org/abs/2309.03882.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.05685.

A. Summary of Benchmark Fragility Evidence

We focus on MMLU (Hendrycks et al., 2021) and GSM8k (Cobbe et al., 2021) because their ubiquity makes them the highest-value targets for contamination and the most extensively studied cases; together they also cover two distinct evaluation modalities, multiple-choice knowledge and free-form mathematical reasoning, ensuring the fragility pattern is not an artifact of a single format. Table 2 consolidates the perturbation findings from Section 2. The perturbations span three qualitatively distinct categories: *positional* (answer reordering), *linguistic* (question rephrasing), and *symbolic* (variable substitution). That substantial degradation occurs across all three categories, both benchmark types, and independent research groups rules out any explanation specific to a single perturbation strategy, benchmark design, or research team. The pattern is systematic.

B. Worked Examples of Generative Procedures

The central practical question raised by dynamic evaluation is what a "generative procedure" concretely looks like. This appendix gives three worked examples spanning distinct modalities: symbolic reasoning, long-context retrieval, and sequential decision-making. Each procedure is a deterministic function of a published random seed s, so that any evaluator who runs it with the same seed and version obtains the same instances (reproducibility), while no evaluator can memorize instances that are only realized when the procedure is run (instance-level contamination resistance). In each case the ground truth is computed by the generator itself, with no human annotation.

Example 1: Symbolic arithmetic reasoning (GSM-style). A template specifies a word problem with typed, named placeholders and a symbolic solution. At evaluation time, concrete values are sampled and the answer is computed by the same symbolic solver that defines the template, following the substitution principle of GSM-Symbolic (Mirzadeh et al., 2025).

```
seed s; rng ← RNG(s)
rate ← rng.int(2, 9)
hours ← rng.int(3, 12)
name ← rng.choice(NAMES)
prompt ← ``{name} earns {rate}
dollars per hour and works {hours}
hours.  How much is earned?''
answer ← rate × hours
```

Difficulty is tuned by the value ranges, the number of reasoning steps in the template, and the number of irrelevant clauses injected; conceptual diversity is increased by drawing from a library of structurally distinct templates rather than a single one.

Example 2: Long-context retrieval (RULER-style). A "needle" fact is embedded at a controlled depth within a long distractor "haystack," and the model is queried for it; ground truth is the inserted fact, known by construction (Hsieh et al., 2024).

```
seed s; rng ← RNG(s)
key ← rng.uuid(); value ←
rng.uuid()
haystack ← rng.sample(FILLER,
length L)
pos ← rng.int(0, L)
context ← insert(haystack, ``{key}:
{value}'', pos)
prompt ← context + ``What is the
value for {key}?''
answer ← value
```

Context length L, needle depth `pos`, and the number of competing distractor keys are all generative parameters, directly controlling difficulty along the long-context axis.

Example 3: Offline decision-making task (procedural gridworld). The generator emits an entire environment instance whose optimal return is computable, in the spirit of procedurally generated RL benchmarks (Cobbe et al., 2020) and parameterized optimization suites (Hansen et al., 2020).

```
seed s; rng ← RNG(s)
grid ← rng.maze(size n,
wall_density ρ)
start, goal ← rng.free_cells(grid,
2)
hazards ← rng.free_cells(grid, k)
env ← GridWorld(grid, start, goal,
hazards)
ground_truth ← value_iteration(env)
```

The agent (or offline-trained policy) is scored against the optimal value obtained by an exact solver. Grid size n, wall density ρ, and hazard count k tune difficulty, and a held-out range of seeds can be reserved to test offline-to-online generalization on instances drawn from the same distribution but never seen during training. The same pattern extends to bandit and continuous-control instances whose optimal value is analytically or numerically available.

Table 2. Benchmark fragility under semantics-preserving perturbations. In each case the correct answer is unchanged; performance drops reveal sensitivity to surface form rather than underlying capability.

Benchmark	Perturbation	Finding	Source
MMLU	Answer choice reordering	$>$10 percentage point accuracy drop; some models show up to 13 pp variation across orderings	(Pezeshkpour & Hruschka, 2023; Zheng et al., 2024)
MMLU	Answer choice reordering (scale)	Effect confirmed across a broad range of model families and sizes	(Gupta et al., 2024)
MMLU	Linguistic rephrasing	Significant accuracy degradation despite identical factual content	(Wang et al., 2024a)
GSM8k	Symbolic substitution (names, values)	Consistent performance drop across all tested models; variance increases with problem complexity	(Mirzadeh et al., 2025)
GSM8k	Numerical, rephrasing, irrelevant clauses	Systematic degradation confirmed across all perturbation types	(Li et al., 2024a)