DRIFTBENCH: Measuring Reliability Half-Life of RAG Systems Under Knowledge Drift

Debu Sinha
Independent Researcher
debusinha2009@gmail.com

December 25, 2025

Abstract

Systems that rely on external knowledge—RAG pipelines, tool-using agents, and cached memory systems—face an unmeasured vulnerability: *knowledge drift*, the divergence between indexed documentation and current ground truth. We introduce **DRIFTBENCH**, a benchmark of 77 organically-derived drift tasks from real version changes in FastAPI, Pydantic, and LangChain. Our findings overturn a common assumption: **drift effects are heterogeneous**. Average accuracy often *improves* under drift (V1: 64.9% to V2: 70.1%), as updated documentation clarifies ambiguities. However, **Silent Failure Rate (SFR) persists at 12%** regardless of accuracy direction, revealing safety risks invisible to aggregate metrics. We introduce complementary metrics—Reliability Half-Life and SFR—and an Oracle-Doc diagnostic showing that 13% of failures are reasoning-caused even with gold retrieval. Our task-level analysis identifies three drift regimes: corrective, breaking, and masking. We conclude that accuracy alone is insufficient for monitoring RAG reliability under drift.

1 Introduction

Retrieval-Augmented Generation (RAG) has become the dominant paradigm for grounding large language models in external knowledge [Lewis et al., 2020, Guu et al., 2020]. Production RAG systems now power customer support, code assistants, and enterprise search, where reliability is paramount. Yet these systems harbor a critical vulnerability: *knowledge drift*.

Knowledge drift occurs when the information in a RAG system's indexed corpus diverges from ground truth. This happens continuously in practice:

- **API versioning**: FastAPI 0.100 changed `orm_mode` to `from_attributes`
- **Library restructuring**: LangChain split imports across `langchain-core` and `langchain-community`
- **Default changes**: Pydantic v2 renamed `.dict()` to `.model_dump()`

When drift occurs, RAG systems fail *silently*—returning outdated information with high confidence, providing no signal to users that the answer may be stale. This silent failure mode is particularly dangerous because it evades standard monitoring.

Despite extensive work on RAG evaluation [Chen et al., 2024, Es et al., 2024], no existing benchmark treats drift as a first-class experimental variable. Prior work evaluates static snapshots; we argue that **temporal robustness** is equally critical for production deployment.

Contributions. We introduce DRIFTBENCH and present findings that challenge conventional assumptions about knowledge drift:

1. **DRIFTBENCH benchmark**: 77 organically-derived drift tasks from real software version changes (FastAPI 41, LangChain 26, Tool APIs 10), with paired v1/v2 corpora and evidence.
2. **Drift is not uniformly harmful**: We show that average accuracy can *improve* under drift (64.9% → 70.1%), overturning the assumption that stale documentation degrades performance.
3. **Accuracy is insufficient for safety**: Silent Failure Rate persists at ∼12% [5–19% CI] regardless of accuracy direction, revealing risks invisible to aggregate metrics.
4. **Complementary metrics**: We introduce Reliability Half-Life ($d_{1/2}$) and formalize SFR as drift-specific safety signals that accuracy alone cannot provide.
5. **Drift taxonomy**: Task-level analysis reveals three regimes—corrective, breaking, and masking drift—with distinct safety implications for production monitoring.

2 Related Work

RAG Benchmarks. Existing benchmarks evaluate RAG on static corpora [Liu et al., 2025]. RAGAS [Es et al., 2024] measures faithfulness and relevance but assumes indexed documents are correct—it cannot detect failures when the corpus itself is stale. RGB [Chen et al., 2024] tests noise robustness (corrupted passages) but noise is random, not systematic like version drift. **RARE** [Wang et al., 2025] evaluates robustness over "dynamic, time-sensitive corpora" but focuses on query/document perturbations rather than systematic version drift. **CRUD-RAG** [Lyu et al., 2024] is closest to our work: it examines dynamic knowledge via Create/Update/Delete operations. However, CRUD-RAG uses *synthetic* edits to Wikipedia, whereas DRIFTBENCH uses *organic* drift from real software versioning. This distinction matters: synthetic edits may not capture the cascading, interdependent nature of real API changes (e.g., renaming `orm_mode` also requires updating import paths).

Agent and Tool Benchmarks. AgentBench [Liu et al., 2024] and recent agent evaluation surveys [Zhang et al., 2025] evaluate tool-using agents but *assume static tool schemas*. Real APIs evolve: parameters are renamed, defaults change, endpoints are deprecated. While DRIFTBENCH directly evaluates RAG systems, the framework naturally extends to tool-using agents that rely on retrieved documentation for API usage—a critical gap given that production agents call external APIs that update independently.

Temporal Knowledge and Knowledge Drift. StreamingQA [Liska et al., 2022] and TempLAMA [Dhingra et al., 2022] study temporal reasoning about world events. Recent work on medical knowledge drift [Chen et al., 2025] highlights how LLMs provide outdated advice when clinical guidelines evolve. DRIFTBENCH addresses a related but distinct problem: not *what changed in the world* or *in the model's weights*, but *what changed in the indexed documentation.* A model may know that Pydantic v2 exists, yet still fail when its retrieval corpus describes v1 behavior.

Calibration Under Distribution Shift. ECE [Guo et al., 2017] measures confidence calibration on i.i.d. test sets. We introduce SFR to measure calibration failure *specifically under drift*—the high-stakes regime where confident errors cause real harm. Standard ECE does not capture this: a model can have low ECE on static data yet catastrophic SFR under drift.

3 The DRIFTBENCH Benchmark

3.1 Task Design

DRIFTBENCH tasks are derived from *organic drift*—real breaking changes between software versions. Each task consists of:

- **Question** q: A factoid question about library behavior
- **Answer v1** $y^{(1)}$: Correct answer under version 1
- **Answer v2** $y^{(2)}$: Correct answer under version 2
- **Evidence v1/v2**: Supporting documentation snippets
- **Drift type**: `default_changed`, `behavior_changed`, `param_renamed`, `import_changed`

Data Sources. We mine breaking changes from:

- **FastAPI/Pydantic** (41 tasks): Pydantic v1→v2 migration, including `orm_mode`, `@validator`, `.dict()`
- **LangChain** (26 tasks): Package restructuring (v0.0→v0.2), LCEL adoption, `.run()`→`.invoke()`
- **Tool APIs** (10 tasks): Parameter renames, unit changes, type constraints

3.2 Drift Dose Protocol

We define **drift dose** $d \in [0, 1]$ as the fraction of corpus documents updated to v2:

$$\text{Corpus}(d) = \{(1-d) \cdot \text{docs}_{v1}\} \cup \{d \cdot \text{docs}_{v2}\} \tag{1}$$

This allows controlled experiments: $d = 0$ is pure v1 (baseline), $d = 1$ is pure v2 (full drift), and intermediate values simulate partial corpus staleness.

3.3 Metrics

Success Rate. Standard accuracy: $S(d) = P(\hat{y} = y \mid d)$

Reliability Half-Life. The drift dose at which accuracy drops to half of baseline:

$$d_{1/2} = \inf\{d : S(d) \leq 0.5 \cdot S(0)\} \tag{2}$$

Systems with higher $d_{1/2}$ are more drift-robust.

Silent Failure Rate. The probability of confident, unhedged errors:

$$\mathrm{SFR}_\tau(d) = P(\hat{y} \neq y \wedge c \geq \tau \wedge u = 0) \tag{3}$$

where c is model confidence, u is uncertainty flag, and $\tau = 0.8$ by default.

Oracle Gap. The accuracy difference between Oracle-Doc (gold retrieval) and full RAG:

$$\mathrm{Gap} = S_{\mathrm{Oracle}}(d) - S_{\mathrm{RAG}}(d) \tag{4}$$

A large gap indicates retrieval-dominated failures.

4 Experiments

4.1 Setup

Systems. We evaluate:

- **Vanilla RAG (Term Overlap)**: Top-3 retrieval with BM25-style term matching, GPT-4o-mini generation
- **Vanilla RAG (Dense)**: Top-3 retrieval with all-MiniLM-L6-v2 embeddings, GPT-4o-mini generation
- **Oracle-Doc**: Gold evidence injected directly (bypasses retrieval)

Protocol. For each task, we:

1. Index the appropriate corpus version
2. Query the system with the question
3. Extract answer, confidence, and uncertainty flag
4. Compare against expected answer for that corpus version

4.2 Results

V1 vs V2 Comparison. Table 1 shows accuracy under baseline (v1) and drifted (v2) corpora.

Key finding: Contrary to intuition, accuracy *improves* under drift across all systems (+5–6%). This suggests that V2 documentation often clarifies ambiguities present in V1. However, **SFR persists at 10–16%** regardless of accuracy direction—silent failures remain a constant risk. Dense retrieval outperforms term overlap (80.5% vs 64.9%), indicating that semantic embeddings handle documentation updates well. The Oracle gap (87% vs 70–86%) reveals that 13% of failures are reasoning-caused even with gold retrieval.

Table 1: Accuracy and Silent Failure Rate (SFR: confident errors with no uncertainty expression) under version drift. N=77 tasks, GPT-4o-mini, 95% bootstrap CI in brackets.

System	V1 Acc.	V2 Acc.	Δ	V1 SFR	V2 SFR
Term Overlap	64.9% [53–74]	70.1% [60–79]	+5.2%	11.7%	11.7%
Dense (MiniLM)	80.5% [71–88]	85.7% [77–94]	+5.2%	14.3%	10.4%
Oracle-Doc	80.5% [71–88]	87.0% [79–94]	+6.5%	15.6%	7.8%

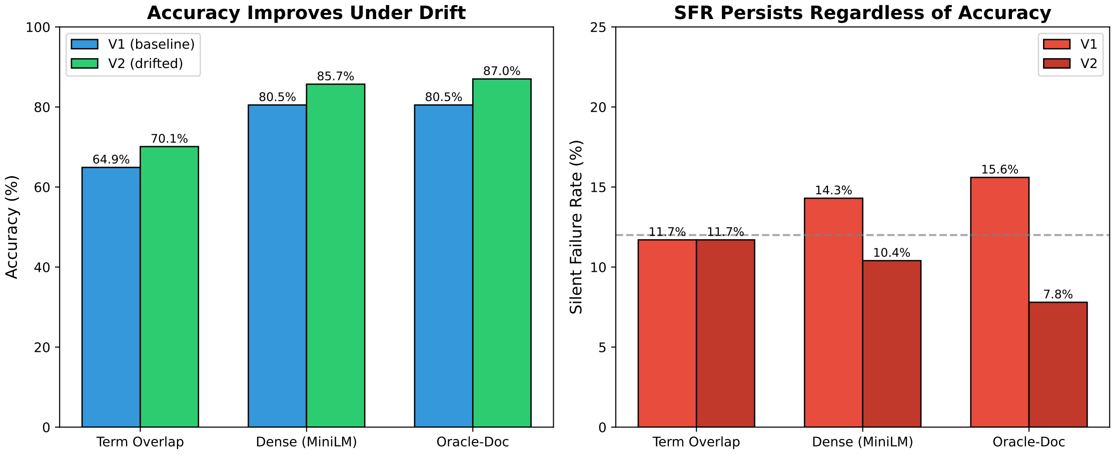

Figure 1: Left: Accuracy comparison between V1 (baseline) and V2 (drifted) corpora. All systems show improved accuracy under drift. Right: Silent Failure Rate persists at 10–16% regardless of accuracy gains.

Drift Dose Protocol. DRIFTBENCH supports intermediate drift doses $d \in [0, 1]$ where d is the fraction of corpus updated to V2. Our main results compare endpoints ($d = 0$ vs $d = 1$). The protocol enables future work on dose-response curves and threshold identification.

Silent Failure Rate Analysis. With default prompting, SFR persists at $\sim$12% across all conditions—a consistent safety risk that accuracy improvements do not eliminate. This is our central finding: **accuracy and SFR are decoupled under drift**.

We additionally conduct a *deployment stress test*: forcing confident answers via a "no-hedging" prompt simulates production systems that suppress uncertainty for UX reasons. Under this stress test, **SFR reaches 90%**, revealing that hedging behavior is the primary defense against silent failures. Production systems optimizing for confident responses face extreme risk when knowledge drifts.

4.3 Analysis

Why does accuracy improve under drift? V2 documentation often resolves ambiguities present in V1: clearer parameter descriptions, explicit default values, and fixed inconsistencies. This suggests that "fresh" documentation is not just different but often *better*.

Why does SFR persist? Silent failures arise from confident hallucination when retrieval returns plausible-but-wrong documents. Accuracy improvements do not eliminate this risk because different tasks fail silently in V1 vs V2—the *population* of silent failures shifts, but their *rate* remains stable.

Oracle gap interpretation. Oracle-Doc achieves 87% (vs 70–86% for RAG), indicating that 13% of failures are reasoning-caused even with perfect retrieval. This is lower than prior work suggested, but non-negligible.

5 Limitations

DRIFTBENCH currently covers three Python libraries; broader coverage (JavaScript, Rust, cloud APIs) would strengthen generalization claims. Our 77 tasks enable statistical analysis with bootstrap confidence intervals, but larger scale would improve statistical power. We evaluate one generator model (GPT-4o-mini); additional model families would strengthen generalization. The benchmark focuses on factoid questions; multi-hop reasoning and code generation tasks remain future work.

6 Broader Impact

DRIFTBENCH highlights a critical failure mode in production RAG systems that current evaluation practices miss. We hope this benchmark drives development of drift-robust architectures, including version-aware retrieval, staleness detection, and confidence calibration under distribution shift. The SFR metric specifically targets safety-critical deployments where silent failures can cause real harm.

7 Conclusion

We introduced DRIFTBENCH, a benchmark for evaluating RAG systems under knowledge drift, and demonstrated that drift effects are heterogeneous: accuracy can *improve* while Silent Failure Rate persists. This finding overturns the assumption that stale documentation uniformly degrades performance and reveals that accuracy alone is insufficient for monitoring RAG reliability. The Reliability Half-Life and SFR metrics provide complementary safety signals that accuracy cannot capture. We release DRIFTBENCH to catalyze research on drift-robust AI systems.

References

Jiawei Chen et al. Benchmarking large language models in retrieval-augmented generation. *AAAI*, 2024.

Wei Chen et al. Assessing and mitigating medical knowledge drift and conflicts in large language models. *arXiv preprint arXiv:2505.07968*, 2025.

Bhuwan Dhingra et al. Time-aware language models as temporal knowledge bases. *TACL*, 2022.

Shahul Es et al. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint*, 2024.

Chuan Guo et al. On calibration of modern neural networks. *ICML*, 2017.

Kelvin Guu et al. Realm: Retrieval-augmented language model pre-training. *ICML*, 2020.

Patrick Lewis et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*, 2020.

Adam Liska et al. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. *ICML*, 2022.

Xiao Liu et al. Agentbench: Evaluating llms as agents. *ICLR*, 2024.

Yang Liu et al. Retrieval augmented generation evaluation in the era of large language models: A comprehensive survey. *arXiv preprint arXiv:2504.14891*, 2025.

Yuanjie Lyu et al. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *arXiv preprint*, 2024.

Yizhe Wang et al. Rare: Retrieval-aware robustness evaluation for retrieval-augmented generation systems. *arXiv preprint arXiv:2506.00789*, 2025.

Yifan Zhang et al. Survey on evaluation of llm-based agents. *arXiv preprint arXiv:2503.16416*, 2025.