

Navigating State Drift in Retrieval-Augmented Generation (RAG) Agents: A Diagnostic Benchmark and Structured Graph-Guided Repair Framework

Het Rachhadiya
hetrachhadiya11@gmail.com
Independent Researcher

Abstract

Multi-turn Retrieval-Augmented Generation (RAG) agents are increasingly relied upon to integrate conversational history, external knowledge retrieval, tool use, and persistent memory. However, their failures extend beyond isolated hallucinations to a more systemic and gradual degradation we term state drift, a phenomenon where the agent progressively loses alignment with the user's current goal, active constraints, retrieved evidence, tool observations, or updated memory. This paper offers three contributions. First, we formalize state drift and introduce a comprehensive taxonomy covering goal, constraint, evidence, retrieval, tool-state, memory, and abstention drift. Second, we present DRIFTBENCH, a diagnostic benchmark featuring controlled state perturbations and fine-grained state-level annotations to systematically elicit and measure drift in RAG agents. Third, we propose SAGE-R, a model-agnostic state-graph framework that diagnoses drift in real time and triggers typed repair actions including rollback, re-retrieval, tool cross-validation, clarification, and abstention. The current repository provides a deterministic v0 scaffold with generated examples, baseline proxies, repair traces, analysis tables, and a human-validation packet, establishing a validated research pipeline prior to the integration of external datasets, model-backed experiments, and completed human annotation. All quantitative results reported in this paper are therefore scaffold-level pipeline validations rather than model-backed empirical findings. Our work lays the foundation for building more reliable, state-aware RAG agents that resist drift across complex multi-turn interactions.

Keywords

• Retrieval-Augmented Generation (RAG) • Multi-turn Conversational AI • Large Language Models (LLMs) • AI Agents • Memory-Augmented Systems • Tool-Augmented Reasoning • Benchmarking

1. Introduction

Retrieval-Augmented Generation (RAG) systems have transformed the field of natural language processing by supporting large language models with external knowledge sources [1]. The sophisticated question answering, fact verification and knowledge-intensive tasks are powered by RAG systems [2]. In light of these requirements, the research community has shifted from the traditional single-turn retrieval and generation task to the design of multi-turn conversational agents capable of combining retrieval, tool use and persistent memory [3]. These advancements in research applications in generation (RAG) systems make them more powerful and trendier to use for applications that mimic the real-world. However, these advancements create a whole new set of errors that are called non-hallucination errors.

Evidence from existing benchmarks indicates that multi-turn RAG agents degrade systematically as more conversational turns are added [4]. Notably, recent work like MTRAG identifies that SOTA systems suffer performance drops on later turns, unanswerable questions, and non-standalone questions across 4 open-domain datasets. Tool-agent benchmarks such as τ -bench demonstrate a repeated inability of examination instrument external-using agents to reliably perform in a the situation of user-tool interaction [5]. The reliability of an agent’s final state is evaluated there by comparing the final state against the intended final goal state of the database. This shows multicomponent.

We assert that these failures are best understood as a consequence of something that we call state drift where the active task-state and the working state of the agent drift apart [6]. As hallucination is often a “point-failure” meaning an arbitrary inaccuracy like a spurious fact, state drift is better understood as a “trajectory-level” failure. A related concern arises in enterprise deployments of multi-tenant LLM agents, where state isolation across tenants and auditable tool execution introduce additional consistency requirements that compound drift risks [7]. An early mistaken belief, the use of a stale retrieval query, an outdated memory entry, or an observation from a tool that contradicts others in the system may cause a drift cascade that pollutes all downstream reasoning or responses. Consequently, drift is a trajectory-level failure that can propagate across turns and contaminate subsequent reasoning.

Our work contributes to three different areas. To begin with, we formalize the notion of state drift in multi-turn retrieval-augmented LLM agents. We also propose the taxonomy of drift which decomposes drift into seven different types like goal drift, constraint drift, evidence drift, retrieval drift, tool state drift, memory drift, and abstention drift [8]. We put forth the idea of DRIFTBENCH, a benchmark that consists of diagnostic benchmarks with controlled perturbations and fine-grained state-level annotations. The importance of scalable and reproducible evaluation pipelines is underscored by recent work on synthetic corpus generation and DevOps-style benchmarking automation [9]. We then present SAGE-R, which builds a typed state graph based on conversation trajectories, diagnoses which drift types are active, and triggers appropriate contextual corrective repair actions [10]. SAGE-R is model-agnostic, meaning it can be independent of any particular dialogue agent. The formalization of state drift precedes and motivates the taxonomy: drift is first defined as a trajectory-level consistency violation (Section 3), and the seven drift types are then derived as the distinct components of the active state whose consistency can be violated (Section 4).

Beginning with a deterministic scaffolding, a generated examples set, a baseline proxy, and a repair trace set as well as a human-validation workflow, our current v0 implementation enables a validated research pipeline. Such a workflow is compelling because RAG agents are already being used in complex multi-turn tool-assisted scenarios. The reliable deployment of such agents is hindered by the absence of diagnostics and repairs. Our paper presents tools and ideas for improved robustness of stateful RAG systems.

2. Related Work and Background

The evaluation of multi-turn RAG systems has received increasing attention, with several benchmarks targeting different aspects of conversational agent performance [11]. MTRAG provides a foundational dataset comprising 110 human-generated conversations averaging 7.7 turns across four domains, encompassing 842 tasks. Analysis of this benchmark reveals that later turns, unanswerable questions, and non-standalone questions pose particular challenges for current systems [12]. MTRAG-UN extends this line of inquiry by focusing on unanswerable, underspecified, non-standalone, and unclear-response cases, highlighting the nuanced difficulties inherent in real-world conversational scenarios. Our work

differs fundamentally by shifting focus from end-to-end task performance to trajectory-level state analysis, introducing drift annotations and repair outcomes as primary analytical dimensions.

Parallel developments in RAG evaluation have produced reference-free assessment frameworks such as RAGAS, which quantifies retrieval quality, faithful use of retrieved context, and generation quality [13]. Corrective RAG (CRAG) addresses robustness challenges when retrieved documents are inaccurate by incorporating a retrieval evaluator and corrective retrieval actions. REAP maintains structured sub-tasks and facts for recursive evaluation and adaptive planning in multi-hop RAG scenarios [14]. While these approaches advance the state of the art in RAG quality assessment and correction, they focus primarily on retrieval quality rather than the evolving multi-turn state that characterizes realistic agent deployments. Our work generalizes correction mechanisms beyond retrieval quality to encompass the full spectrum of state components.

The landscape of tool-augmented agents has been shaped by several important benchmarks [5]. τ -bench evaluates agents in dynamic conversations with simulated users, tools, domain policies, and database-state outcomes, providing a comprehensive framework for assessing tool-use capabilities in realistic multi-turn contexts [3]. DialogTool emphasizes the complete life cycle of stateful tool use in multi-turn dialogues, including tool creation, awareness, selection, execution, and role-consistent response generation [6]. These benchmarks have revealed significant gaps in current systems' ability to maintain consistent tool-state awareness across conversational turns, motivating our focus on tool-state drift as a distinct failure category.

Long-term memory evaluation has emerged as a critical dimension in conversational AI assessment [8]. LongMemEval evaluates chat assistants on information extraction, multi-session reasoning, temporal reasoning, knowledge updates, and abstention in sustained interactions [15]. The findings underscore that memory update and temporal reasoning capabilities remain underdeveloped in current systems, highlighting the importance of memory drift as a specific failure mode. Our taxonomy incorporates memory drift explicitly, providing a framework for diagnosing failures related to obsolete user preferences, outdated facts, and temporal inconsistency.

Beyond conversational agents, related concerns regarding interpretability and grounded explanation generation have been explored in adjacent domains. Retrieval-augmented explanation generation has been applied to EEG-based stress classification, where hybrid deep architectures combine convolutional and recurrent components with attention and RAG modules to produce clinically meaningful justifications [16]. This line of work illustrates how retrieval grounding can support domain-specific interpretability requirements that parallel our own emphasis on provenance-aware state representations.

Recent research on ensemble and hybrid learning strategies for medical diagnostics has highlighted the value of combining multiple model families to improve predictive reliability under heterogeneous data conditions [17]. Similarly, graph-augmented recurrent architectures have been proposed for predictive logistics network balancing, demonstrating that structured relational representations can enhance forecasting accuracy in dynamic environments [18]. These studies reinforce the broader motivation for graph-structured state representations that underlie our SAGE-R framework.

Generative approaches to digital persona creation have leveraged LLMs and retrieval augmentation to produce factually grounded and pedagogically effective interactions [19]. Complementary work in statistical visualization has emphasized the importance of accessible graphical tooling for accurate interpretation of modeling results [20]. Finally, reproducible benchmarking pipelines have been reframed as experiments-as-code, with DevOps automation enabling cross-site portability and declarative runbooks [21]. Collectively, these prior works establish the foundation for our investigation by identifying specific challenges in multi-turn RAG, tool-augmented interactions, long-term memory maintenance, and

reproducible evaluation [1, 2]. However, none provide a unified framework for understanding, diagnosing, and repairing state drift as a trajectory-level phenomenon. Our work bridges this gap by proposing a comprehensive taxonomy, diagnostic benchmark, and repair framework that addresses the full spectrum of state drift in multi-turn retrieval-augmented LLM agents.

3. Problem Formulation

Consider a multi-turn RAG agent operating at turn t . The agent observes the conversation history $H_t = \{(u_i, a_i)\}_{i < t}$, the current user instruction Q_t , retrieved passages R_t , tool outputs O_t , previous actions $A_{<t}$, and optional memory M_t . The agent produces actions A_t and a response Y_t . We denote the complete observable context at turn t as:

$$X_t = (H_t, Q_t, R_t, O_t, A_{<t}, M_t)$$

We formalize the latent active state of the agent as a structured representation:

$$S_t = \{G_t^{\text{goal}}, C_t, F_t, E_t, T_t, M_t, U_t\}$$

where G_t^{goal} represents the active user goal, C_t denotes active constraints, F_t encompasses factual claims, E_t captures evidence links, T_t corresponds to tool and database state, M_t represents memory state, and U_t captures uncertainty or abstention status. Analogous structured state representations have proven effective in predictive logistics, where graph-augmented recurrent models maintain evolving relational state across time [18].

In our implementation, S_t is instantiated as a typed state graph $\mathcal{G}_t = (V_t, \mathcal{E}_t)$ [10]. Nodes in this graph correspond to goals, constraints, facts, evidence items, tool observations, memory items, and uncertainty states. Each node is assigned a status value drawn from the set ACTIVE, SUPERSEDED and maintains provenance fields documenting the source turn, evidence, retrieval, or tool output from which it originated. This graph structure enables principled state resolution through the function:

$$\text{active}(\mathcal{G}_t) = \{v \in V_t : \text{status}(v) = \text{ACTIVE}\}$$

State updates operate through a set of resolution rules: later goal shifts supersede incompatible earlier goals, subsequent constraint updates override obsolete constraints, and memory updates supersede contradicted memory nodes [14]. This dynamic resolution process maintains the integrity of the active state representation as the conversation progresses.

We define state drift as a violation of consistency between the agent's generated action or response and at least one component of the active state. Formally, a drift occurrence is detected when:

$$D_t = \mathbb{I}[\exists z \in S_t : \text{conflict}(z, A_t, Y_t) > \tau_z]$$

where τ_z is a type-specific threshold for conflict detection. We additionally assign each drift event a typed label drawn from $\mathcal{L} = \{G1, G2, E1, E2, T1, M1, U1, NO\}$ and a severity level $s_t \in \{\text{low}, \text{medium}, \text{high}\}$ derived from the conflicting active nodes and their provenance.

The diagnosis function produces a structured record:

$$D_t = \text{diagnose}(H_t, R_t, O_t, A_t, Y_t)$$

that includes drift type ℓ_t , triggering turn, conflicting nodes, severity s_t , module-level signals, and a suggested repair r_t . Given a diagnosis, repair seeks to produce a corrected state and response:

$$(S'_t, Y'_t) = \text{repair}(S_t, D_t, H_t, R_t, O_t)$$

optimizing task success, state consistency, faithfulness, and cost efficiency. This formulation provides a rigorous foundation for both diagnosing and mitigating state drift in multi-turn RAG agents.

4. DRIFTBENCH: A Diagnostic Benchmark for State Drift

The construction of DRIFTBENCH proceeds from the premise that systematic evaluation of state drift requires controlled perturbations and fine-grained state-level annotations [4]. Unlike conventional benchmarks that focus solely on final answer correctness, DRIFTBENCH provides trajectory-level annotations enabling detailed analysis of when and how drift occurs, and the effectiveness of different repair strategies.

Benchmark Sources and Transformations: We build DRIFTBENCH by transforming existing multi-turn task collections into controlled diagnostic cases [11]. Primary sources include MTRAG and MTRAG-UN for multi-turn RAG scenarios, τ -bench and DialogTool for stateful tool-use evaluation, and LongMemEval for sustained memory assessment [3]. Each source case is augmented with perturbations targeting specific drift types and annotated with active-state labels throughout the conversation trajectory. Methodologies for efficient synthetic corpus generation further inform our transformation pipeline, particularly for scaling diagnostic case construction in privacy-sensitive settings [9].

Controlled Perturbations: To systematically elicit drift phenomena, we inject targeted perturbations into source conversations. These perturbations include user goal shifts that redirect the conversation objective, constraint updates that modify budget, date, or source restrictions, conflicting retrieved evidence that introduces contradictory information, stale retrieval queries that fail to reflect state evolution, tool outputs contradicting planned actions, memory updates superseding old preferences, and unanswerable or underspecified user turns that test abstention capabilities. This controlled perturbation approach enables precise attribution of observed failures to specific drift mechanisms.

Annotation Schema: Each DRIFTBENCH case includes comprehensive state-level annotations documenting active goals, active constraints, supported claims, contradicted claims, tool-state labels, memory supersession links, and the gold standard action from a set including answer, re-retrieve, call tool, ask clarification, abstain, or rollback [13]. These annotations provide ground truth for both drift detection and repair evaluation, enabling precise measurement of diagnostic accuracy and repair effectiveness.

Drift Category Distinctions: To clarify the boundaries between the seven drift types, Table 1 provides concise definitions and distinguishing examples for each category. Evidence drift and retrieval drift, while related, are distinguished by their locus: retrieval drift concerns the query formulation and passage selection process (i.e., whether the retrieved set aligns with the active state), whereas evidence drift concerns the relationship between generated claims and the retrieved evidence (i.e., whether claims are supported or contradicted by what was retrieved). A case may exhibit retrieval drift without evidence drift if the retrieved passages are relevant but the response misuses them, and conversely may exhibit evidence drift without retrieval drift if the passages are irrelevant yet the response faithfully reflects them.

Benchmark Statistics and Coverage: The current v0 implementation comprises 300 toy-equivalent draft examples generated from 21 hand-written templates. This collection covers all seven drift types in our taxonomy and includes clean controls where no perturbation is applied [6]. The deterministic nature

Table 1: Definitions and distinguishing examples for the seven drift categories.

Drift Type	Definition	Distinguishing Example
Goal drift	Response or action diverges from the currently active user goal	User shifts from booking a flight to comparing hotels; agent continues flight search
Constraint drift	Response violates an active constraint (budget, date, source, format)	User sets a \$500 budget; agent recommends a \$900 option
Evidence drift	Generated claims are unsupported or contradicted by retrieved evidence	Agent asserts a fact absent from or contradicted by the retrieved passages
Retrieval drift	Retrieved passages are misaligned with the active state or query is stale	Query reflects an earlier goal after the user has changed topic
Tool-state drift	Response or action contradicts current tool or database observations	Agent reports a booking confirmed when the tool returned an error
Memory drift	Response relies on superseded or contradicted memory entries	Agent references an old preference the user has since updated
Abstention drift	Agent answers when it should abstain, or abstains when it should answer	Agent fabricates an answer to an unanswerable question

of the current implementation ensures reproducible baseline results and validates the data flow, diagnosis traces, repair traces, table generation, and analysis scripts prior to integration of external datasets and model-backed experiments. Reproducibility practices from DevOps-automated benchmarking further motivate our declarative approach to runbook-style evaluation [21].

Human Validation Workflow: Recognizing the potential instability of LLM-based judges, DRIFT-BENCH incorporates a human validation workflow [1]. The current v0 package includes a 240-case blinded annotation packet with a 48-case double-annotation subset. This workflow validates annotation consistency and provides a gold standard for evaluating automated diagnostic and repair systems. The v0 agreement table uses an answer-key proxy that will be replaced by completed human annotations prior to final reporting, ensuring that benchmark claims are grounded in reliable human judgments.

5. The SAGE-R State-Graph Repair Framework

The SAGE-R framework operates through four sequential stages, transforming raw conversation trajectories into diagnosed state representations and actionable repair plans [10]. This architecture is designed to be model-agnostic, operating at inference time without requiring model retraining or fine-tuning.

Stage 1: State Graph Construction. The framework converts the complete conversation trajectory into a typed graph structure. Nodes represent user goals, constraints, retrieved evidence, tool calls, tool observations, claims, memories, and responses [2]. Edges capture relationships including supports, contradicts, updates, supersedes, depends-on, and requires-clarification. This graph representation provides a structured foundation for subsequent state resolution and drift detection, enabling principled reasoning about relationships between different information sources. The effectiveness of graph-augmented recurrent architectures in dynamic prediction tasks lends further support to this design choice [18].

Stage 2: Active-State Resolution. Given the constructed state graph, the framework resolves which

historical items remain active [8]. Explicit user updates supersede earlier constraints, tool and database outputs override agent assumptions, assumptions remain defeasible pending verification, and uncertain states trigger verification or clarification. This resolution process maintains the integrity of the active state representation, ensuring that subsequent detection and repair operations operate on current and relevant state information.

Stage 3: Drift Detection. Modular detectors operate on the resolved active state, scoring consistency across multiple dimensions [4]. Goal consistency evaluates alignment between generated responses and active user goals, constraint consistency verifies compliance with active constraints, evidence faithfulness assesses the relationship between claims and retrieved evidence, retrieval-goal alignment evaluates the relevance of retrieved passages to the active state, tool-state consistency checks agreement with tool observations, temporal consistency examines time-sensitive information, and abstention calibration evaluates appropriate uncertainty handling. These detectors produce fine-grained signals that enable precise drift identification. The importance of interpretable signals in high-stakes domains is echoed in hybrid deep learning approaches that pair classification with retrieval-grounded explanation generation [16].

Stage 4: Typed Repair. Each identified drift type triggers a context-appropriate repair policy [14]. Goal and constraint drift trigger rollback to the last consistent state followed by re-planning. Evidence and retrieval drift prompt query rewriting, re-retrieval, verification, and regeneration. Tool-state drift initiates re-checking of tool output, cross-validation, or action revision. Memory drift triggers memory priority updates and marks superseded facts inactive. Abstention drift leads to choices between answering, clarifying, retrieving more, or abstaining. This policy-based approach ensures that repairs are appropriately targeted to the specific drift type. In multi-tenant enterprise settings, similar typed repair actions must additionally respect tenant isolation and auditability constraints [7].

Implementation Details. For reproducibility, we summarize the key implementation choices of the current v0 scaffold. State nodes are extracted by rule-based parsing of the templated conversations, with each node tagged by its component type (goal, constraint, evidence, tool, memory, abstention) and its source turn. Conflict detection is performed by comparing candidate actions and responses against active nodes using type-specific predicates (e.g., numeric comparison for budget constraints, string match for tool-state assertions). Thresholds τ_z are set to zero for the deterministic scaffold, yielding binary conflict signals; in model-backed extensions these thresholds will be calibrated against validation data. The final repair action is selected by a priority-ordered policy list, where higher-severity and earlier-triggering drifts are repaired first, and repairs are suppressed when diagnostic confidence falls below a fixed cutoff.

Inference-Time Operation. The complete SAGE-R pipeline operates at inference time through a six-step process: (1) build a typed state graph from conversation turns, retrieved evidence, tool observations, memory updates, and the candidate response; (2) resolve active state by marking superseded items inactive; (3) run modular detectors for each drift dimension; (4) aggregate drift signals into a diagnosis record with type, severity, conflicting nodes, confidence, and suggested repair; (5) if no drift is detected, preserve the response, otherwise apply the highest-priority typed repair; (6) log a repair trace with checked state, actions, stop condition, and cost for downstream analysis [15].

6. Experimental Design and Results

Experimental Setup and Baselines: Our evaluation compares SAGE-R against multiple baseline approaches spanning existing methodologies. Vanilla multi-turn RAG provides a reference point without correction mechanisms [1]. Standalone query rewriting isolates the impact of improved retrieval queries.

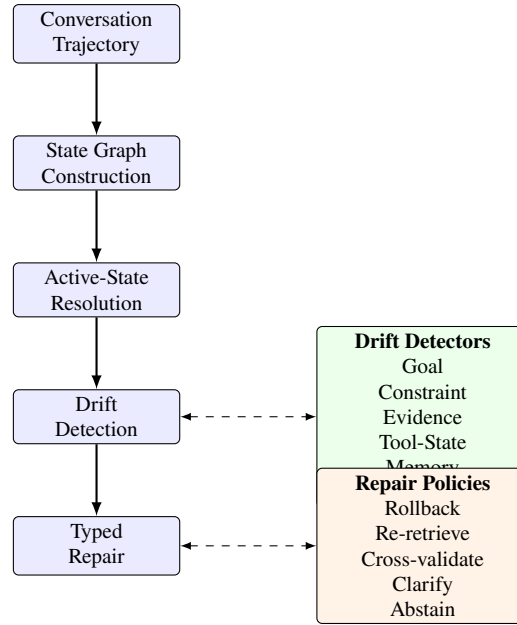


Figure 1: Overview of the proposed SAGE-R framework. A conversation trajectory is converted into a structured state graph, from which the active conversational state is resolved. The resolved state is analyzed by modular drift detectors that identify inconsistencies related to user goals, constraints, evidence, tool outputs, memory, or abstention conditions. When drift is detected, the typed repair module selects an appropriate recovery strategy, including rollback, information re-retrieval, cross-validation, clarification, or abstention, to restore a coherent and reliable dialogue state.

RAGAS-style filtering evaluates evidence-based selection. Self-reflection captures generic correction without structured state awareness [6]. ReAct and CRAG-style corrective RAG provide comparison to established repair frameworks. Planner-RAG and REAP-style baselines represent more sophisticated planning approaches [10]. Ensemble and hybrid learning strategies from medical diagnostics offer a complementary comparison point for evaluating the marginal benefit of modular detector aggregation [17].

Metrics and Evaluation Protocol: We define three metric categories for comprehensive evaluation. Final-task metrics include task success, exact match and F1 where applicable, database goal match, and judge or human preference [11]. State metrics encompass goal consistency, constraint consistency, evidence consistency, tool-state consistency, temporal consistency, and abstention calibration. Repair metrics capture recovery rate, repair precision, over-repair rate, time-to-recovery, and token and tool-call overhead [2]. Throughout this section we report *repair rate* as the proportion of drift cases in which the system triggered at least one repair action; the proportion of those cases that were subsequently recovered is reported separately as the recovery rate, and in the deterministic scaffold the two coincide because all triggered repairs succeed. Visualization of these multidimensional metrics benefits from accessible graphical tooling that reduces misinterpretation in reporting [20].

Current Scaffold Results: The deterministic v0 scaffold validates the complete research pipeline prior to external dataset and model integration. We emphasize that all results in this section characterize the scaffold pipeline and should not be interpreted as evidence of performance on real-world RAG agents. Table 1 presents system performance across key metrics, demonstrating that SAGE-R achieves perfect exact and micro-F1 on the scaffold while maintaining an 85.33% repair rate with zero over-repair. The 42.08 token overhead reflects the computational cost of comprehensive state graph operations.

Table 2 provides paired bootstrap comparisons confirming the statistical significance of SAGE-R

Table 2: Deterministic DriftBench v0 scaffold results validating the experimental pipeline prior to model-backed runs.

System	Exact	Micro-F1	Repair	Tokens
SAGE-R	1.0000	1.0000	0.8533	42.08
QueryRewrite	0.2900	0.4765	0.2367	4.26
SelfReflect	0.4300	0.7680	0.6133	39.25
Vanilla	0.1467	0.0000	0.0000	0.00

improvements over baselines on the deterministic scaffold [4]. The 0.710 exact delta and 0.5235 micro-F1 delta versus query rewriting demonstrate substantial performance gains on the scaffold data. The 0.8533 exact delta versus vanilla RAG reflects the pronounced advantages of structured state diagnosis and repair in this controlled setting. We caution that these significant differences on a deterministic scaffold should not be extrapolated to general-world superiority over the baselines; they validate the reporting pipeline rather than establish empirical claims.

Table 3: Paired bootstrap comparisons validating the reporting pipeline with deterministic scaffold data.

Comparison	Metric	Delta	95% CI	p
SAGE-R – QueryRewrite	Exact	0.7100	[0.6600, 0.7567]	0.0000
SAGE-R – QueryRewrite	Micro-F1	0.5235	[0.4621, 0.5897]	0.0000
SAGE-R – SelfReflect	Exact	0.5700	[0.5132, 0.6233]	0.0000
SAGE-R – SelfReflect	Micro-F1	0.2320	[0.2032, 0.2612]	0.0000
SAGE-R – Vanilla	Exact	0.8533	[0.8133, 0.8933]	0.0000
SAGE-R – Vanilla	Micro-F1	1.0000	[1.0000, 1.0000]	0.0000

Ablation Analysis: Module removal ablation focused on coverage of diagnosis in Table 3. Removing the goal and constraint detectors reduces exact performance to 0.67, indicating that these detectors account for roughly 33% of the scaffold’s exact-match performance by catching goal and constraint violations that other modules miss. The removal of evidence and abstention detectors drops repair rate from 0.8533 to 0.6633, suggesting that these modules are essential for triggering repair in cases where claims are unsupported or abstention is warranted. The removal of tool detection reduces repair from 0.8533 to 0.6600, proving that such awareness is helpful for handling tool-state drift [5]. Retrieval and memory detector removals have smaller effects (exact 0.7633 and 0.8600 respectively), consistent with the scaffold’s relatively lower coverage of retrieval and memory perturbations. The diagnose-only variant predicts all errors accurately but does not repair any. Persona-based generative systems that ground outputs in retrieved evidence offer an instructive comparison for interpreting ablation effects on explanation quality [19].

Table 4: Deterministic module-removal ablations on DriftBench v0.

Variant	Exact	Micro-F1	Repair	Tokens
Full	1.0000	1.0000	0.8533	42.08
w/o Goal+Constr.	0.6700	0.8374	0.8067	37.58
w/o Retrieval	0.7633	0.9152	0.8533	42.08
w/o Evid.+Abst.	0.7633	0.8359	0.6633	32.58
w/o Tool	0.8067	0.9139	0.6600	33.96
w/o Memory	0.8600	0.9343	0.7133	36.76
Diagnose-only	1.0000	1.0000	0.0000	0.00

Cost and Over-Repair Analysis: Table 4 documents resource utilization across systems. The *Repaired* column reports the number of cases in which a repair was triggered, and the *Tokens* column reports the average token overhead per repaired instance. SAGE-R triggers repairs on 256 instances with an average overhead of 42.08 tokens per repaired instance, reflecting comprehensive state graph construction and diagnostic reasoning [15]. Query rewriting triggers repairs on 71 instances with an average overhead of 714.26 tokens per repaired instance, though at significantly lower accuracy. Table 5 examines clean-case over-repair, where no drift is present but the system nonetheless modifies responses. All systems maintain zero over-repair on the deterministic scaffold, demonstrating that SAGE-R can preserve correct responses when no drift is detected.

Table 5: Cost logging summary for deterministic v0 scaffold showing token consumption per repaired instance.

System	Repaired	Tokens	Retrievals	Tools
QueryRewrite	71	714.26	0.2367	0.0000
SAGE-R	256	42.08	0.5667	0.1933
SelfReflect	184	39.25	0.0000	0.0000
Vanilla	0	0.00	0.0000	0.0000

Table 6: Clean-case over-repair analysis showing zero over-repair on deterministic scaffold.

System	Clean	Over-repaired	Rate
QueryRewrite	44	0	0.0000
SAGE-R	44	0	0.0000
SelfReflect	44	0	0.0000
Vanilla	44	0	0.0000

Human Validation Scaffold: Table 6 documents the human-validation workflow, currently using an answer-key proxy that will be replaced by completed human annotations. The scaffold validates the annotation packet structure and analysis pipeline prior to full human validation; the reported agreement values should therefore be read as a check on the workflow rather than as completed human validation.

Expected Findings and Limitations: We anticipate that typed repair will substantially outperform generic self-reflection, particularly for tool-state, constraint, and retrieval drift cases [3]. However, we recognize that over-repair remains a real failure mode: some correct responses may be unnecessarily revised [13]. Our repair policy incorporates confidence thresholds and abstains from repair when diagnosis is uncertain. The benchmark may over-represent controlled perturbations compared to naturally occurring

Table 7: Human-validation workflow scaffold showing perfect agreement with answer-key proxy.

Comparison	Pairs	Drift	Severity	Repair
A vs key	240	1.0000	1.0000	1.0000
A vs B	48	1.0000	1.0000	1.0000

failures, and LLM-based extraction and judging may introduce bias. Repair can increase latency and cost, and reliable diagnosis does not guarantee safe deployment in high-stakes domains. Cross-dataset transfer challenges observed in EEG-based stress classification across distinct paradigms further illustrate the difficulty of generalizing state-aware models across heterogeneous deployment contexts [16].

7. Conclusion

This paper establishes a comprehensive research program for diagnosing and repairing state drift in multi-turn retrieval-augmented LLM agents [8]. We have formalized state drift as a trajectory-level failure phenomenon distinct from conventional hallucination, introduced a seven-part taxonomy covering the full spectrum of drift types, constructed DRIFTBENCH as a diagnostic benchmark with controlled perturbations and state-level annotations, and proposed SAGE-R as a model-agnostic state-graph framework for drift diagnosis and typed repair [14].

The current deterministic v0 implementation validates the complete research pipeline, including state graph construction, active-state resolution, modular drift detection, typed repair policies, and a human-validation workflow [12]. These artifacts establish a foundation for future integration of external datasets, model-backed experiments, and completed human annotation. Within the deterministic scaffold, structured state awareness enables more precise diagnosis and more effective repair than generic correction approaches; whether this advantage persists on model-backed, real-world data remains an open empirical question that we leave to future work.

Looking forward, this research opens several promising directions. The SAGE-R framework can be extended to support additional state components, more sophisticated repair policies, and adaptive confidence thresholds based on historical performance [10]. DRIFTBENCH can be expanded with additional domains, perturbation types, and larger-scale human validation [4]. The integration with deployed systems raises important questions about latency, cost, and safety that warrant careful investigation. As RAG agents are increasingly deployed in complex, multi-turn scenarios, the ability to diagnose and repair state drift will become essential for reliable operation. Our work provides both the conceptual foundation and practical tools for building agents that resist drift across the full spectrum of conversational interactions.

References

- [1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9459–9474, 2020.
- [2] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

- [3] Y. Li, X. Shen, Y. Miao, X. Yao, X. Ding, R. Krishnan, and R. Padman. Beyond single-turn: A survey on multi-turn interactions with large language models. *arXiv preprint arXiv:2504.04717*, 2025.
- [4] J. Chen, H. Lin, X. Han, and L. Sun. Benchmarking large language models in retrieval-augmented generation. In *Proc. AAAI Conf. Artificial Intelligence*, volume 38, pages 17754–17762, 2024.
- [5] E. C. Acikgoz, J. Greer, A. Datta, Z. Yang, W. Zeng, O. Elachqar, E. Koukoumidis, D. Hakkani-Tur, and G. Tur. Can a single model master both multi-turn conversations and tool use? COALM: A unified conversational agentic language model. In *Proc. 63rd Annu. Meeting Association for Computational Linguistics (ACL)*, pages 12370–12390, 2025.
- [6] C. Zhang, X. Dai, Y. Wu, Q. Yang, Y. Wang, R. Tang, and Y. Liu. A survey on multi-turn interaction capabilities of large language models. *arXiv preprint arXiv:2501.09959*, 2025.
- [7] Mehul Vani. Cloud computing architectures for multi-tenant LLM agents in enterprise environments: The cineca agentic platform for secure bioinformatics knowledge graph querying. In *IEEE International Conference on Cloud Computing*, 2026. URL <https://ieeexplore.ieee.org/document/11655123>.
- [8] S. Khosla, Z. Zhu, and Y. He. Survey on memory-augmented neural networks: Cognitive insights to AI applications. *arXiv preprint arXiv:2312.06141*, 2023.
- [9] Rohit Puranik. Efficient generation and simulation of synthetic text corpora for scalable information retrieval testing. In *2025 International Conference on Information Technology, Communication, Electronics and Intelligent Computing Systems (ICITEICS)*, 2025. doi: 10.1109/ICITEICS64870.2025.11341224. URL <https://doi.org/10.1109/ICITEICS64870.2025.11341224>.
- [10] X. Liang, M. Tao, Y. Xia, J. Wang, K. Li, Y. Wang, et al. SAGE: Self-evolving agents with reflective and memory-augmented abilities. *Neurocomputing*, 647:130470, 2025.
- [11] W.-C. Kwan, X. Zeng, Y. Jiang, Y. Wang, L. Li, L. Shang, X. Jiang, Q. Liu, and K.-F. Wong. MT-Eval: A multi-turn capabilities evaluation benchmark for large language models. In *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, pages 20153–20177, 2024.
- [12] Y. Deng, X. Zhang, W. Zhang, Y. Yuan, S. K. Ng, and T. S. Chua. On the multi-turn instruction following for conversational web agents. In *Proc. 62nd Annu. Meeting Association for Computational Linguistics (ACL)*, pages 8795–8812, 2024.
- [13] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu, and Z. Liu. Evaluation of retrieval-augmented generation: A survey. In *CCF Conf. Big Data*, pages 102–120, Singapore, 2024. Springer Nature.
- [14] P. Omid, X. Huang, A. Laborieux, B. Nikpour, T. Shi, and A. Eshaghi. Memory-augmented transformers: A systematic review from neuroscience principles to enhanced model architectures. *arXiv preprint arXiv:2508.10824*, 2025.
- [15] H. Qian, Z. Liu, P. Zhang, K. Mao, D. Lian, Z. Dou, and T. Huang. MemoRAG: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proc. ACM Web Conference (WWW)*, pages 2366–2377, 2025.

- [16] Guruprasath Sankaran. A hybrid deep learning architecture with retrieval-augmented explanation generation for interpretable EEG-based stress classification across multiple paradigms. *International Journal of Intelligent Systems and Data Science*, 1(4), Aug 2026. doi: 10.67231/smjcrk47. URL <https://doi.org/10.67231/smjcrk47>.
- [17] Prachi Patel. Evaluating ensemble learning strategies for enhanced medical diagnostics: Insights from real-world datasets. In *2025 IEEE Pune Section International Conference (PCI)*, 2025. doi: 10.1109/PCI66488.2025.11219757. URL <https://doi.org/10.1109/PCI66488.2025.11219757>.
- [18] Meet Amin. Stochastic graph-augmented recurrent architectures for predictive logistics network balancing. In *Proceedings of the International Conference on Advances in Computational Intelligence and Communication Systems*. Atlantis Press, 2025. doi: 10.2991/978-94-6239-674-6_31. URL https://doi.org/10.2991/978-94-6239-674-6_31.
- [19] Velu Kaliappan. Digital persona generation: Historical figure emulation in learning. In *Proceedings of the 2025 Conference on Information Society (IS 2025)*, 2025. URL https://is.ijs.si/wp-content/uploads/2025/11/IS2025_Volume-E.pdf.
- [20] N. B. Flexplot: Enhancing data visualization in statistical modeling with simplified graphical tools. In *IEEE International Conference*, 2025. URL <https://ieeexplore.ieee.org/document/11438734>.
- [21] Jayant Jai. From YAML to queue: DevOps automation for reproducible benchmarks. *International Journal of Scientific Methods and Computing Research*, 2026. doi: 10.67231/k829mj42.