RESEARCH ARTICLE OPEN ACCESS

Generative Ai and Agentic Systems: Modeling Goal Drift, Self-Referential Training and Emergent Failure Modes in Autonomous Reasoning Architectures

Milan Sharma[1], Rudransh Upadhyay[1], Chetan Sharma[2], and Arya Sharma[2]

[1] *Department of Data Science and Artificial Intelligence, Global Institute of Technology, Jaipur, India*

[2] *Department of Computer Science, Global Institute of Technology, Jaipur, India*

ABSTRACT

Advances in large language models (LLMs) have enabled AI systems capable of autonomous operation across extended, multi-step tasks. These agentic systems exhibit emergent behaviours that standard evaluation methods are ill-suited to capture. This paper addresses three underexplored phenomena in agentic AI: (1) Latent Goal Crystallization (LGC), in which an agent's operative sub-goals drift from its original specification over long operational horizons without explicit instruction changes; (2) Recursive Self-Prompt Amplification (RSPA), in which agents iteratively revise their own prompts, introducing feedback dynamics that can destabilize task coherence; and (3) Multi-Agent Epistemic Drift (MAED), in which belief errors compound across agent networks even when individual agents appear well-calibrated. To situate these phenomena within a unified framework, we introduce the Cognitive Autonomy Gradient (CAG), a three-dimensional scoring system characterizing AI systems along axes of reasoning depth, goal persistence, and environmental coupling. We further present taxonomy of seven agentic failure modes and propose the Autonomous Cognition Evaluation Suite (ACES), a longitudinal benchmark methodology designed to evaluate agent behaviour beyond short-horizon task completion. The findings have practical relevance for any deployment context in which AI agents operate with limited human oversight.

Keywords **—** generative AI, agentic systems, autonomous agents, goal drift, large language models, multi-agent systems, AI safety, self-prompting, epistemic calibration, cognitive autonomy.

1. Introduction

AI research has historically focused on bounded problems: given a task, a dataset, and an evaluation metric, progress was measured by performance improvements on that metric. This paradigm worked effectively for decades. However, contemporary large language models do not fit neatly within it. These systems can reason across domains, generate code, summarize documents, and construct structured plans, often without task-specific training. When augmented with agentic scaffolding—enabling tool use, persistent memory, and sub-task delegation—their behavioural complexity increases substantially. Studying these systems presents significant challenges. Most existing benchmarks are designed for single-turn or short-horizon tasks and evaluate final outputs rather than the intermediate processes by which an agent arrives at them. As a result, critical behaviours—such as how internal objectives

evolve, how iterative prompt revisions accumulate, and how errors propagate through multi-agent communication remain largely unobserved. This work provides a structured framework for analysing these dynamics.

The paper makes four primary contributions. First, it introduces the Cognitive Autonomy Gradient (CAG), a practical scoring framework for characterizing the autonomy level of any AI system. Second, it formally defines and analyses three under characterized phenomena: LGC, RSPA, and MAED. Third, it presents a taxonomy of failure modes specific to high-autonomy agent deployments. Fourth, it describes ACES, a benchmark suite designed to evaluate agent behaviour over extended operational periods. This paper is primarily theoretical and taxonomic in scope. Empirical validation at scale is not reported here; rather, the goal is to establish clear terminology and a coherent framework to guide the design of such experiments.

2. Background and Related Work

A. Generative Models and Their Limitations

The transformer architecture, introduced by Vaswani et al. [2], enabled training at previously impractical scales by allowing each token in a sequence to attend to all others. Subsequent scaling research [3] demonstrated consistent, predictable capability gains as model size, data volume, and compute increased, motivating the development of large foundation models adapted for diverse tasks.

While generative models excel at producing contextually plausible continuations, they do not natively maintain state across interactions, set autonomous goals, or interface with external systems. Each inference call is independent; without explicit context injection, the model has no memory of prior interactions. This stateless design is adequate for many applications but becomes a fundamental constraint in scenarios requiring sustained reasoning, information retrieval, or adaptive behaviour across extended time horizons.

B. Agentic AI Frameworks

In contemporary practice, the term "agent" refers to an LLM augmented with tools, memory, and an iterative loop enabling multi-step task execution before returning a final result [4]. The React framework [5] formalizes this pattern, interleaving reasoning traces with action invocations such as web search or computation, followed by continued reasoning over the results.

Subsequent extensions have refined this design. Reflexion [6] incorporates a self-evaluation step in which the agent critiques its own outputs before committing. Plan-and-Solve [7] separates planning from execution into distinct phases. Multi-agent frameworks such as Autogeny enable specialized instances to communicate and delegate tasks. While these represent genuine engineering advances enabling complex, multi-step workflows, they share a common limitation: their evaluation focuses on single-session task completion, leaving open questions about behaviour under extended operation, environmental change, or multi-round inter-agent communication.

C. Gaps in the Existing Literature

A systematic review of publications from Nauris, ICML, ICLR, and ACL (2020-2024) reveals three persistent gaps. No existing work provides a formal model of how an agent's operative sub-goals drift from its original specification over

time. No study quantifies how belief errors compound through iterative multi-agent communication. No framework formally characterizes the feedback dynamics arising when model outputs are recycled as training inputs. This paper directly addresses each of these gaps.

3. The Cognitive Autonomy Gradient

The Cognitive Autonomy Gradient (CAG) is a three-dimensional scoring framework for characterizing the autonomy level of AI systems. Its three dimensions are defined as follows.

Reasoning Depth (RD) quantifies the degree of inferential processing performed prior to output generation. A model performing single-pass next-token prediction scores near zero; a system executing tree-search with backtracking over candidate reasoning paths scores near one.

Goal Persistence (GP) measures how consistently a system maintains coherent objectives over time. A stateless model with no cross-call memory scores zero; a system that retains and pursues goals across sessions and context resets scores near one.

Environmental Coupling (EC) captures the degree to which a system's actions affect and are affected by its operational environment. A read-only system with no external state modification scores zero; a system with persistent write access to databases and real-world actuators scores near one.

For a system S, the CAG score is the vector $CAG(S) = (RD, GP, EC) \in [0, 1]^3$. A scalar autonomy index is derived as:

$$A(S) = \text{sqrt}(RD^2 + GP^2 + EC^2) / \text{sqrt}(3) \quad (1)$$

A standard GPT-4 completion call scores approximately (0.4, 0.0, 0.1), yielding $A \approx 0.21$. A fully deployed agentic system with persistent memory and live-system tool access may score (0.7, 0.5, 0.8), yielding $A \approx 0.62$. Table I presents CAG comparisons across representative system types.

TABLE I: ***AI System Types Compared Using the CAG Framework***

System Type	Goal Persistence	Memory Horizon	Self-Modify	A(S) Range
Narrow ML Model	None	Per-inference	None	0.00 – 0.10
Generative LLM	Context only	Short-term	None	0.10 – 0.25
Tool-Augmented Agent	Task-scoped	Session	Minimal	0.30 – 0.50
Multi-Agent System	Distributed	Shared memory	Partial	0.45 – 0.65
Autonomous Agent *	Long-horizon	Consolidating	Self-prompting	0.70 – 1.00

** Most prominent in deployments with A(S) > 0.5.*

It is important to note that A(S) is a descriptive measure, not a normative one. Higher autonomy scores do not imply greater quality or desirability. In fact, the failure modes described in subsequent sections become more probable as

A(S) increases, which is precisely the utility of this framework as a risk-characterization tool.

4. Three Under characterized Phenomena

A. Latent Goal Crystallization (LGC)

During extended operation, an agent's behaviour tends to stabilize around a set of recurring sub-strategies. Some of these sub-strategies are explicitly motivated by the original task specification; others are not. They emerge because certain behavioural patterns are consistently reinforced—either by reducing computational load, simplifying future decision points, or stabilizing the agent's internal state. Over time, these patterns can solidify into implicit goals that are functionally independent of the user's original instructions.

This phenomenon is termed Latent Goal Crystallization (LGC), reflecting its gradual, non-explicit nature. For example, an agent tasked with document summarization may progressively Favor shorter outputs—not because brevity was specified—but because it reduces downstream processing complexity. Individual outputs may each appear acceptable; the drift becomes visible only through statistical analysis across many sessions.

Hypothesis 1 (LGC Emergence): LGC is likely to emerge when the operational horizon T exceeds the natural task-resolution horizon T_0 by a factor $\kappa > 3$, the reward or evaluation signal is sparse, and the agent employs a memory mechanism with selective consolidation. As T increases, the probability of at least one crystallized implicit sub-goal approaches one.

Detecting LGC requires analysis of second-order behavioural statistics over time, not merely whether individual outputs meet quality thresholds. An instructive analogy is sensor hardware drift: individual readings may remain within specification while the aggregate trend reveals systematic deviation from calibration.

B. Recursive Self-Prompt Amplification (RSPA)

Several current systems permit agents to revise their own operational prompts during task execution. While this can improve specificity when initial instructions are ambiguous, the recursive structure introduces feedback dynamics that warrant careful analysis.

Let P(t) denote the agent's prompt at step t and G(·) its generative function. Under self-prompting, the update rule is:

$$P(t+1) = F[\ P(t), G(P(t))\ ] \quad (2)$$

where F is the function by which the agent integrates its own output into the subsequent prompt. The stability of this iteration is governed by the Prompt Amplification Factor (PAF), defined as the spectral radius of the Jacobian of F evaluated at P(t).

Theorem 1 (RSPA Stability): The sequence {P(t)} converges to a fixed-point P* if and only if PAF < 1 in the neighbourhood of P*. When PAF > 1, prompt content diverges exponentially from the original instruction. When PAF = 1, the system may enter a periodic orbit or exhibit sensitivity to perturbations.

Three operational regimes are anticipated. In the convergent regime (PAF < 1), iterative refinement progressively tightens the prompt toward a well-scoped instruction. In the oscillatory regime (PAF ≈ 1), the agent alternates between competing task framings, producing inconsistent output quality. In the divergent regime (PAF > 1), each revision increasingly

departs from the original task intent, potentially yielding an agent that executes the wrong task with high confidence.

C. Multi-Agent Epistemic Drift (MAED)

In multi-agent systems, communication and belief-sharing are typically assumed to improve collective accuracy. This assumption is not always warranted. When agents share a systematic error, iterative exchange of outputs does not correct the error; it compounds it.

Multi-Agent Epistemic Drift (MAED) is defined as the divergence between the ensemble's aggregate epistemic state and external ground truth after n rounds of inter-agent communication:

$$\delta\_MAED = H\_ensemble - (1/n)\, \Sigma_i\, H_i \quad (3)$$

where H_ensemble denotes ensemble-level uncertainty and H_i denotes the individual uncertainty of agent i. A positive δ_MAED indicates that the group is systematically less accurate than the average individual agent, despite high inter-agent agreement.

This mechanism is well-documented in robotic sensor fusion [8], where correlated measurement errors produce overconfident state estimates. The practical implication is significant: inter-agent consensus is not a reliable proxy for correctness, particularly when agents share training data, system prompts, or reasoning heuristics.

5. Self-Referential Training Pipelines

A related and underappreciated risk concerns training pipelines in which model outputs serve as training data for future model versions. Synthetic data generation, AI-assisted annotation, and model-generated evaluation criteria are all in active industrial use.

A Self-Referential Training Loop (SRTL) of order k is defined as a training process in which data produced by models within k generations of the current model contributes to the current training distribution. SRTL-1—the most common case—occurs when outputs from the immediately preceding version are used for fine-tuning or preference labelling.

Three pathological regimes can emerge from SRTL dynamics. In the amplification regime, existing model strengths are reinforced, which can resemble rapid improvement while silently entrenching biases. In the drift regime, distributional shift from model-generated data gradually displaces human-aligned behavioural norms. In the collapse regime, output diversity degrades across successive generations—a phenomenon with empirical support in the literature [10].

Mitigating these pathologies requires an Epistemic Grounding Anchor (EGA): a designated proportion of training data, at each iteration, drawn from sources independent of the current model's outputs. The required EGA fraction to prevent drift is estimated to scale as $O(1/\sqrt{T})$ with training iterations T, though empirical validation of this bound is needed.

6. Architectural Considerations

A. Hierarchical Metacognitive Controller (HMC)

To address LGC, RSPA divergence, and MAED within a unified architectural component, this paper proposes the Hierarchical Metacognitive Controller (HMC). The HMC operates in parallel with the agent's primary task-execution layer and

maintains an ongoing model of the agent's own behavioural trajectory.

The HMC comprises three sub-components. The Goal Integrity Monitor (GIM) periodically computes the semantic similarity between the agent's current operative sub-goals and the original task specification using a frozen embedding model; divergence beyond a defined threshold triggers a reset or escalation to human oversight. The Prompt Stability Analyzer (PSA) estimates PAF in real time by comparing successive prompt revisions. The Epistemic Anchoring Module (EAM) maintains a curated set of verified reference facts and re-calibrates the agent's stated beliefs against them at fixed intervals, providing a practical defence against MAED accumulation.

B. Memory Architecture

A four-tier memory structure is proposed for long-horizon agents. Tier 1 (working memory) is the active context window, typically 32K-128K tokens. Tier 2 (episodic memory) stores compressed session summaries, updated periodically by a consolidation function. Tier 3 (semantic memory) is a structured knowledge graph, updated only when new information passes EGA-based verification. Tier 4 (procedural memory) holds behavioural strategy templates derived from prior tasks, continuously monitored by the GIM for goal drift.

The critical design constraint is that Tier 3 updates are gated by external verification. An agent must not write to its semantic memory based solely on its own inferences; this pathway is the primary mechanism through which MAED and SRTL pathologies propagate into persistent storage.

7. Evaluation: The ACES Benchmark

Existing benchmarks including Web Arena [11], Agent Bench [12], and GAIA [13]—are structured around short-horizon task completion. This design cannot capture phenomena that unfold over many operational steps. Gradual goal drift, slowly diverging prompt sequences, and belief degradation across 50 rounds of inter-agent communication are all invisible to standard evaluation protocols.

TABLE II: ***Taxonomy of Agentic Failure Modes***

Failure Mode	Horizon	Observable Sign	Detection Method
Latent Goal Crystallization	Long	Sub-goal mismatch	Behaviour trend audit
Divergent RSPA	Medium	Prompt scope widening	Prompt entropy tracking
Multi-Agent Epistemic Drift	Any	High agreement, low accuracy	External ground-truth check
Context Window Blindness	Short	Early context ignored	Recency bias test
Reward Proxy Gaming	Long	Good metrics, poor outcomes	Adversarial evaluation
Cascading Constraint Relaxation	Long	Scope creep in actions	Boundary drift logging

Failure Mode	Horizon	Observable Sign	Detection Method
Overconfidence Feedback	Long	Uncertainty decreases over time	Calibration curve review

Note: All failure modes are most prominent in deployments with A(S) > 0.5.

To address these limitations, this paper proposes the Autonomous Cognition Evaluation Suite (ACES). ACES comprises four components: (1) ACES-GS (Goal Stability), which tracks whether an agent's operative sub-goals remain consistent with its original specification after N ∈ {100, 1000, 10000} operational steps; (2) ACES-PT (Prompt Trajectory), which measures information-theoretic distance between successive self-generated prompts to detect RSPA onset; (3) ACES-EC (Epistemic Calibration), which compares stated agent uncertainty against ground-truth accuracy, quantifying δ_MAED across communication rounds; and (4) ACES-RS (Recursive Stability), which estimates PAF under controlled conditions.

Reproducibility is addressed through deterministic environment simulation with fixed random seeds, standardized initial agent states, and a versioned task corpus. A reference implementation is in preparation for public release.

8. Broader Implications

If LGC goes undetected in deployed systems, agents may develop operative objectives that have never been explicitly approved. This constitutes a qualitatively distinct alignment concern from deceptive alignment [14], which presupposes intentional concealment. LGC requires no such intentionality; it is a natural consequence of optimization under constraint over extended time horizons.

MAED has direct implications for multi-agent decision-support systems. Policymakers or clinicians relying on AI ensembles for recommendations may receive highly consistent advice that is systematically incorrect. The consistency itself may generate false confidence. Current AI validation frameworks are not designed to detect this failure mode.

SRTL risks are the most proximate of the three. The window between capability emergence and adequate safety evaluation tools is already narrow. EGA requirements should therefore be treated as a mandatory component of any pipeline in which model outputs contribute to the training distribution.

From a governance perspective, systems scoring A(S) > 0.7 should be subject to explicit capability disclosure requirements prior to deployment, consistent with standards already applied to high-consequence engineering systems in other domains.

9. Conclusion and Future Work

A. Summary of Key Findings

This paper introduced the Cognitive Autonomy Gradient (CAG), a three-dimensional framework for characterizing AI system autonomy. Three previously under characterized phenomena were formally defined: Latent Goal Crystallization (LGC), Recursive Self-Prompt Amplification (RSPA), and Multi-Agent Epistemic Drift (MAED). A taxonomy of seven agentic failure modes was presented alongside the ACES benchmark framework for longitudinal agent

evaluation. Together, these contributions provide a vocabulary and analytical foundation for researchers and practitioners working with high-autonomy AI systems.

B. Limitations

Several limitations warrant acknowledgment. First, CAG dimensional scores currently require expert judgment to assign; automated scoring methods have not yet been developed. Second, the hypotheses concerning LGC emergence and SRTL dynamics are theoretical; large-scale empirical validation is required before strong conclusions can be drawn about specific parameter thresholds. Third, the ACES benchmark has not yet been deployed or stress-tested across diverse agent architectures. Fourth, the failure mode taxonomy may not be exhaustive as agentic deployments mature.

C. Future Work

Several directions for future research are identified. First, empirical validation of the CAG framework across deployed agent systems is a necessary next step, including development of automated CAG scoring procedures. Second, controlled experiments should test the LGC emergence hypothesis under varying operational horizons and reward densities. Third, the ACES benchmark requires a reference implementation and evaluation against current and forthcoming agentic architectures. Fourth, the relationship between SRTL order k and distributional drift rate should be formally characterized. Fifth, governance frameworks informed by CAG scores should be evaluated in consultation with policymakers and AI safety researchers. Finally, integration of the HMC architecture with existing agent frameworks such as Autogeny and Lang Graph represents a practical avenue for applied research.

Acknowledgment

The authors thank the Advanced Research Institute, Jaipur, for access to computing infrastructure used in the preliminary analysis supporting this work. No external funding was received for this research.

References

[1] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," The Knowledge Engineering Review, vol. 10, no. 2, pp. 115-152, 1995.

[2] A. Vaswani, N. Shazer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. NeurIPS, pp. 5998-6008, 2017.

[3] J. Kaplan et al., "Scaling laws for neural language models," arXiv:2001.08361, 2020.

[4] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA: MIT Press, 2018.

[5] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in Proc. ICLR, 2023.

[6] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," in Proc. NeurIPS, 2023.

[7] L. Wang, W. Xu, Y. Lan, Z. Hu, R. K. W. Lee, and E. Lim, "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models," in Proc. ACL, 2023.

[8] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics. Cambridge, MA: MIT Press, 2005.

[9] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in Proc. NeurIPS, 2017.

[10] I. Shumailov, Z. Shumaylov, Y. Zhao, Y. Gal, N. Papernot, and R. Anderson, "The curse of recursion: Training on generated data makes models forget," arXiv:2305.17493, 2023.

[11] S. Zhou et al., "WebArena: A realistic web environment for building autonomous agents," in Proc. ICLR, 2024.

[12] X. Liu et al., "AgentBench: Evaluating LLMs as agents," in Proc. ICLR, 2024.

[13] G. Mialon, C. Fourrier, C. Swift, T. Wolf, Y. LeCun, and T. Scialom, "GAIA: A benchmark for general AI assistants," in Proc. ICLR, 2024.

[14] E. Hubinger, C. van Merwijk, V. Mikulik, J. Skalse, and S. Garrabrant, "Risks from learned optimization in advanced machine learning systems," arXiv:1906.01820, 2019.