Protocol Governance: The Structural Framework for Context-Driven AI Systems

Author: Cody Shelton Affiliation: Independent Researcher, Tri-Arch Research Group

Date: November 23, 2025

Researcher Verification:

ORCID: https://orcid.org/0009-0000-8254-5255

OSF Project: https://doi.org/10.17605/OSF.IO/CN5ZJ

GitHub: Here

Abstract

Recent work such as Stanford's Agentic Context Engineering (ACE) highlights the growing importance of context evolution in large language models (Author et al., 2025)[3]. ACE demonstrates that context can function as an adaptive asset. However, current research offers limited theoretical treatment of how evolving contexts should be structured, governed, or stabilized over time—particularly outside the domain of agentic systems, where ACE's contributions are primarily situated (Author et al., 2025)[3].

This paper proposes Protocol Governance as a conceptual framework for understanding the structural forces that shape context-driven AI behavior, building on CACIM (Shelton, 2024)[1] and CSIP (Shelton, 2025)[2]. We introduce a taxonomy of protocol categories—including grounding protocols, structural protocols, drift-management protocols, reflection protocols, and context-integrity protocols—that serve as foundational organizing principles for stable, interpretable model behavior. These protocols are framed not as prescriptive rules, but as theoretical constructs that describe how models could process, transform, and maintain context under dynamic conditions (Bommasani et al., 2021)[14].

Our aim is not to present empirical validation, but to articulate the theoretical architecture required for such validation to be meaningful. By situating Protocol Governance as a structural layer above existing context-evolution approaches, we establish a foundation on which future research can build formal tests, quantitative metrics, and cross-model evaluations (Kaplan et al., 2020)[15]. This work seeks to provide the conceptual scaffolding

necessary for a systematic study of context stability and drift mechanics in modern AI systems (OpenAI, 2023)[11].

1. Introduction: The New Context Frontier

Large language models (LLMs) increasingly operate as context-driven systems, where behavior, reasoning quality, and apparent coherence emerge not from fixed internal states but from the structure and evolution of the surrounding context (Bommasani et al., 2021)[14]. Recent advances—most notably Stanford's Agentic Context Engineering (ACE)—have begun to explore this frontier by demonstrating that context can function as an adaptive asset (Author et al., 2025)[3]. ACE introduces mechanisms for strategically manipulating context in agentic workflows and shows that context evolution can shape model performance across iterative tasks (Wang et al., 2023)[5].

Yet despite these advances, the broader theoretical landscape remains underdeveloped. Current work provides limited treatment of how evolving contexts themselves should be governed. As models rely more heavily on dynamic, multi-step context accumulation, new challenges arise: context drift, instability under abstraction load, loss of interpretability, and degradation in task alignment (Ganguli et al., 2023)[12]; (Bowman, 2023)[13]. These patterns suggest that context is not merely an informational substrate but a system that behaves according to structural forces—forces that current methodology does not fully describe (Anthropic, 2023)[10]; (OpenAI, 2023)[11].

Context-driven systems raise a foundational question: What governs context? Existing research primarily focuses on prompting, retrieval systems, memory components, and agentic orchestration. These approaches influence context indirectly through careful design or retrieval strategies. However, they do not define the structural principles that determine how context should be formed, maintained, or stabilized over time. Even ACE, while impactful, is situated largely within the domain of agentic system design and does not attempt to articulate a general theory of context governance (Author et al., 2025)[3].

This gap motivates the framework introduced in this paper. We propose Protocol Governance as a conceptual architecture for understanding the structural layer that shapes context-driven AI behavior. Unlike prompting or policy-based methods, protocol governance does not prescribe specific actions to a model. Instead, it outlines theoretical constructs—the protocols—that govern how models process, transform, and sustain context in dynamic, multi-step reasoning environments (Shelton, 2024)[1]; (Shelton, 2025)[2]. These constructs serve as foundational organizing principles for stable, interpretable model behavior (Bommasani et al., 2021)[14].

By framing protocols as the organizing structures behind evolving context, we aim to provide a foundation for systematic study. The goal is not to present empirical validation, but to establish the conceptual scaffolding required for future research into drift mechanics, stability under abstraction, and cross-model consistency (Kaplan et al., 2020)[15]. This framework positions protocol governance as the structural layer that enables context-driven AI systems to remain stable, interpretable, and aligned—even as their contexts become increasingly dynamic and complex (Anthropic, 2023)[10].

2. The Problem: Context Collapse and Cognitive Drift

As LLMs increasingly rely on extended, multi-step context assemblies, a pattern of instability emerges across systems, tasks, and architectures (Bowman, 2023)[13]. These instabilities are often attributed to prompt design flaws or retrieval issues, but closer analysis suggests that they represent deeper structural phenomena that arise from the behavior of evolving contexts themselves (Liu et al., 2023)[6]. While context is frequently treated as a neutral carrier of information, empirical observation across diverse workflows indicates that it exhibits predictable modes of degradation—modes that current approaches do not formally describe (Ganguli et al., 2023)[12].

Context collapse refers to the loss of structure, relevance, or task alignment within multi-step contexts. Collapse can occur gradually as irrelevant information accumulates, or abruptly when prior structure becomes overloaded by abstraction or noise (Anthropic, 2023)[10]. In agentic systems explored by ACE, collapse often manifests when evolving instructions interfere with earlier goals (Author et al., 2025)[3]. Outside agentic settings, collapse appears in tasks that involve long analytic chains, multi-turn reasoning, or iterative refinement (Wei et al., 2022)[4]; (Zhou et al., 2023)[7]. In all cases, the failure mode points toward missing principles for maintaining context integrity over time (Bommasani et al., 2021)[14].

Closely related is cognitive drift, a phenomenon in which the model's behavior diverges from its prior trajectory despite no explicit change in user intent. Drift may present as subtle abstraction shifts, loss of grounding, alteration of role or tone, or inconsistencies in reasoning approach (Bowman, 2023)[13]. Drift is not random; it reflects directional changes in how the model interprets or reprioritizes information (Ganguli et al., 2023)[12]. These changes frequently correlate with shifts in context density, the introduction of high-abstraction material, or recursive references to prior model outputs (Wei et al., 2022)[4].

Both context collapse and cognitive drift share a common structural origin: they arise from the absence of a governing architecture that regulates how context evolves. Current methodologies—prompt engineering, retrieval augmentation, memory systems, and agentic orchestration—address context indirectly (Liu et al., 2023)[6]; (Lewis et al., 2020)[8]. They shape inputs or retrieval strategies, but they do not articulate principles governing how context should maintain coherence as it grows. Without such principles, even well-designed systems exhibit predictable failure modes: gradual erosion of task alignment; compounding abstraction leading to interpretive ambiguity; conflicting contextual cues competing for priority; instability during recursive reasoning; deterioration of internal role consistency; and difficulty recovering from drift once initiated (Anthropic, 2023)[10]; (OpenAI, 2023)[11].

These failures highlight a critical gap in current theory: context-driven AI systems lack a structural model that explains and constrains how context should behave over time (Bommasani et al., 2021)[14]. This paper argues that to understand and eventually mitigate these phenomena, we must first define the structural forces that shape context evolution. The remainder of this work introduces Protocol Governance as a conceptual framework for describing these forces—offering the theoretical constructs needed to study, quantify, and eventually stabilize the behavior of context-driven systems (Shelton, 2024)[1]; (Shelton, 2025)[2].

3. Why Context Alone Isn’t Enough

The growing reliance on extended context has led many researchers to treat context as the primary substrate of reasoning in modern LLMs (Bommasani et al., 2021)[14]. In this view, if context can be shaped, curated, or expanded effectively, model behavior will follow. Approaches such as prompt engineering, retrieval augmentation, memory mechanisms, and agentic orchestration all reflect this assumption: that improving context is synonymous with improving performance (Liu et al., 2023)[6]; (Lewis et al., 2020)[8]; (Wang et al., 2023)[5].

However, this perspective overlooks a critical distinction: context is content, not structure. While context determines what information a model receives, it does not specify how that information should persist, transform, or maintain coherence over time (Anthropic, 2023)[10]. As contexts become longer, more dynamic, and more interdependent, the limitations of treating context as a neutral carrier become increasingly apparent (OpenAI, 2023)[11].

Current methodologies approach context through three primary lenses:

Prompt engineering: shapes the initial state of the context but does not define how it should evolve across multi-step reasoning (Liu et al., 2023)[6].

Retrieval and memory systems: introduce relevant information but do not articulate principles governing its long-term integration or stability (Lewis et al., 2020)[8]; (Borgeaud et al., 2022)[9].

Agentic workflows: manipulate context adaptively, but primarily for coordination and action-taking tasks rather than for establishing a generalizable structural framework (Wang et al., 2023)[5].

These methods influence inputs to the context but do not address the deeper forces that govern context behavior itself. As a result, even well-constructed contexts remain vulnerable to uncontrolled drift, interpretive ambiguity, inconsistent role adherence, abstraction overload, interference between competing instructions, and difficulty recovering from accumulated noise (Ganguli et al., 2023)[12]; (Bowman, 2023)[13]. These limitations stem from a simple but often overlooked point: context has no inherent mechanism for self-regulation (Anthropic, 2023)[10].

This leads to a broader insight: context-driven AI systems require a structural layer that defines how context should behave, not just what it contains. Such a layer would not replace context, but provide the organizing architecture through which context evolves. It would articulate structural principles—conceptual rather than implementation-specific—that shape coherence, stability, and interpretability over time (Shelton, 2024)[1]; (Shelton, 2025)[2]. It would serve as the missing foundation for addressing drift dynamics and collapse patterns identified in the prior section (Bommasani et al., 2021)[14].

The remainder of this paper introduces Protocol Governance as a theoretical framework for defining that structural layer. Protocols do not prescribe specific actions or outputs; rather, they provide principles and patterns that govern how contexts are interpreted, maintained, and stabilized throughout multi-step reasoning processes (Wei et al., 2022)[4]; (Zhou et al., 2023)[7].

4. Protocols: The Missing Framework

The limitations identified in the previous sections point to a structural absence in contemporary AI methodology: while context shapes model behavior, nothing defines how context itself should be shaped, maintained, or governed (Bommasani et al., 2021)[14]. The field lacks a conceptual architecture that explains the structural forces underlying context evolution. Without such an architecture, even advanced prompting strategies, retrieval systems, or agentic workflows can inadvertently amplify drift, collapse, or interpretive instability (Liu et al., 2023)[6]; (Lewis et al., 2020)[8]; (Wang et al., 2023)[5].

This gap motivates the introduction of Protocol Governance. Protocols, in this framework, are conceptual organizing principles that provide structure to evolving contexts. They are not algorithms, prompts, or mechanistic rules. Instead, they serve as theoretical constructs that describe how context maintains coherence, relevance, and interpretability as it changes over time (Shelton, 2024)[1]; (Shelton, 2025)[2].

Just as grammatical rules provide structure to language, protocols provide structure to context evolution. They define the shape, boundaries, and coherence conditions that allow context-driven systems to function reliably across extended reasoning (Wei et al., 2022)[4]. Protocol Governance introduces five primary protocol categories: grounding protocols; structural protocols; drift-management protocols; reflection protocols; and context-integrity protocols (Shelton, 2024)[1]; (Shelton, 2025)[2].

These protocol categories do not control models internally. They do not imply agency, optimization, or goal direction. Instead, they offer language for describing the structural behavior of contexts as they evolve. In other words, they do not tell the model what to do—they tell researchers and designers how to understand what contexts are doing (Bommasani et al., 2021)[14].

By articulating protocols as conceptual constraints, Protocol Governance fills the gap between high-level prompting strategies and empirical behaviors identified in drift and collapse patterns. It creates the theoretical basis for analyzing, predicting, and eventually stabilizing context behavior across a wide range of systems and architectures (OpenAI, 2023)[11]; (Kaplan et al., 2020)[15].

5. A Taxonomy of Protocol Patterns

To operationalize the idea of Protocol Governance, this section introduces a conceptual taxonomy of protocol categories. These categories are not implementation steps or model behaviors; they are structural abstractions that describe the forces influencing context formation, interpretation, and stability. Each category addresses a distinct dimension of how context evolves in multi-turn, multi-layer reasoning (Wei et al., 2022)[4]; (Zhou et al., 2023)[7].

This taxonomy provides the vocabulary needed to analyze context behavior systematically and consistently (Bommasani et al., 2021)[14].

5.1 Grounding Protocols

What they govern: the connection between evolving context and the user's original intent, domain boundaries, and task constraints (Shelton, 2024)[1].

Key questions addressed: How does context remain anchored to its originating purpose? How are task boundaries preserved as context grows? How is relevance maintained across multiple turns? (Liu et al., 2023)[6].

Why it matters: without grounding forces, context becomes susceptible to abstraction drift, competing interpretive frames, and loss of task coherence (Ganguli et al., 2023)[12]. Grounding protocols conceptually describe how to keep context tied to the user's goals even as the interaction becomes more complex (Shelton, 2025)[2].

5.2 Structural Protocols

What they govern: the organizational form of reasoning, including segmentation, ordering, interpretive boundaries, and stepwise structure (Wei et al., 2022)[4].

Key questions addressed: what defines the "shape" of multi-step reasoning? how are steps separated, linked, or prioritized? how do interpretive boundaries remain intact during context evolution? (Zhou et al., 2023)[7].

Why it matters: without structural forces, multi-step reasoning collapses into blended or ambiguous forms. Structural protocols prevent cohesion drift by preserving the relationships between context components (Bommasani et al., 2021)[14].

5.3 Drift-Management Protocols

What they govern: how deviations in interpretive trajectory—directional drift, abstraction drift, recursive drift, or compressive drift—are conceptually identified and constrained (Ganguli et al., 2023)[12].

Key questions addressed: what happens when the reasoning trajectory begins to skew toward one perspective? how does abstraction overload alter interpretive stability? how do recursive references compound drift? (Bowman, 2023)[13]; (Wei et al., 2022)[4].

Why it matters: drift management provides the conceptual framework for identifying when extended reasoning is moving away from task alignment or structural integrity (Shelton, 2024)[1].

5.4 Reflection Protocols

What they govern: structural principles for evaluating, refining, or reorganizing prior reasoning steps within context (Wei et al., 2022)[4].

Key questions addressed: how should prior outputs be reconsidered in multi-step reasoning? how does refinement maintain continuity rather than amplify drift? how does iterative reinterpretation avoid recursive collapse? (Zhou et al., 2023)[7].

Why it matters: reflection is structurally necessary for extended reasoning, but without constraints it can generate recursive drift or abstraction overload. Reflection protocols provide structure without implying internal cognitive processes (Ganguli et al., 2023)[12].

5.5 Context-Integrity Protocols

What they govern: conditions under which context remains coherent, balanced, and interpretable as it accumulates information (Anthropic, 2023)[10].

Key questions addressed: how much context can evolve before segmentation or reorganization becomes necessary? when does accumulated abstraction place structural pressure on interpretability? how do high-density contexts interfere with relevance boundaries? (OpenAI, 2023)[11].

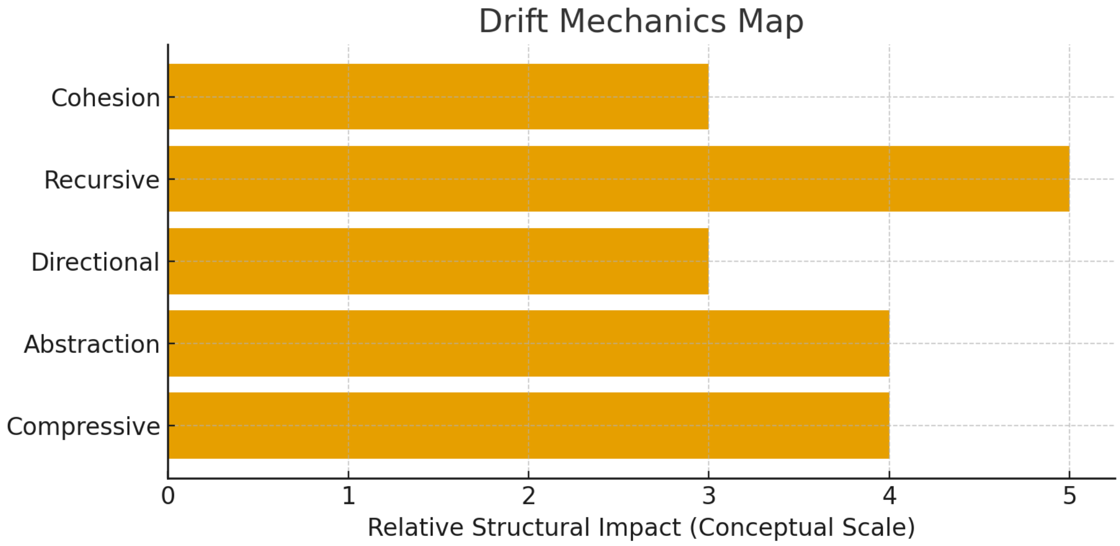
Why it matters: as contexts grow, they accumulate complexity. Integrity protocols define the conditions under which this growth remains manageable and interpretable, preventing collapse under abstraction load (Kaplan et al., 2020)[15].

Closing note on the taxonomy: these five categories are not exhaustive or mechanistic. They serve as conceptual scaffolding for analyzing context behavior. The taxonomy is designed to be flexible, architecture-agnostic, and compatible with future empirical or theoretical refinement (Bommasani et al., 2021)[14].

6. Drift Mechanics: A Structural View

Understanding the behavior of context-driven systems requires more than identifying points of failure; it requires a structural vocabulary for describing how and why contexts deviate from their intended trajectories. Drift is not a malfunction or an anomaly. It is a structural behavior that emerges as context evolves under increasing density, abstraction, or recursive reinterpretation (Bowman, 2023)[13].

This section introduces five principal drift modes. Each mode is conceptual, not mechanistic, and applies across architectures and system designs (Ganguli et al., 2023)[12].



6.1 Compressive Drift

Definition: a drift mode in which distinctions within the context become compressed or blurred as information density increases.

Illustration: a prompt requesting “summarize only the financial impacts” gradually begins producing summaries of general risks as more unrelated content accumulates.

Protocol relevance: grounding and context-integrity protocols help conceptually anchor distinctions, preventing them from being compressed by abstraction load (Anthropic, 2023)[10].

6.2 Abstraction Drift

Definition: a shift from concrete, task-specific details toward broader thematic or generalized statements.

Illustration: a list of specific constraints becomes reinterpreted as “general considerations” after several rounds of refinement or summarization.

Protocol relevance: grounding and reflection protocols maintain alignment between detail-level reasoning and high-level synthesis (Wei et al., 2022)[4]; (Zhou et al., 2023)[7].

6.3 Directional Drift

Definition: a drift pattern in which reasoning increasingly gravitates toward one interpretive direction, lens, or emphasis—even when the task requires balance.

Illustration: a discussion of “benefits and drawbacks” begins disproportionately emphasizing drawbacks after a few negative-leaning iterations.

Protocol relevance: drift-management protocols conceptually define when rebalancing is required to maintain interpretive symmetry (Ganguli et al., 2023)[12].

6.4 Recursive Drift

Definition: a compounding drift pattern triggered when the model interprets (or summarizes) its own prior outputs repeatedly, leading to exponential abstraction or narrowing.

Illustration: summarizing a summary of a summary produces a highly abstract, structurally distorted interpretation detached from the original content.

Protocol relevance: reflection protocols identify conceptual checkpoints where reinterpretation must maintain continuity rather than amplify distortion (Wei et al., 2022)[4].

6.5 Cohesion Drift

Definition: a drift mode in which relationships between contextual components weaken, even if individual components remain relevant.

Illustration: all prior constraints, goals, and assumptions remain in context, yet the model treats them as isolated instead of interconnected.

Protocol relevance: structural and context-integrity protocols preserve the linkages that give context its coherence (Bommasani et al., 2021)[14].

Drift as structural behavior: drift should not be interpreted as a failure of model capability or alignment. Instead, drift arises from unregulated context evolution. These modes reflect predictable structural pressures acting on expanding, multi-turn contexts: abstraction load increases; recursive reinterpretation compounds; interpretive symmetry decays; segmentation boundaries erode; relevance boundaries weaken (Anthropic, 2023)[10]; (OpenAI, 2023)[11]. Drift is therefore a contextual phenomenon, not a cognitive one (Bowman, 2023)[13]. Recognizing drift patterns conceptually is a prerequisite for developing metrics, formal models, or stabilization strategies. Protocol Governance provides the structural vocabulary through which drift can be described, analyzed, and eventually governed (Shelton, 2024)[1].

7. Protocols and Drift: Structural Interactions

Protocols and drift patterns do not exist in isolation. They form a structural ecosystem in which protocol forces shape the stability of context, and drift patterns reveal where those forces weaken or fail to apply (Bommasani et al., 2021)[14]. Understanding the interplay between them is essential for describing how context behaves during extended reasoning (Wei et al., 2022)[4].

7.1 Grounding Protocols and Drift Prevention

Grounding protocols conceptually anchor context to the user's original intent and task boundaries. They counteract abstraction drift by maintaining specificity, directional drift by preserving interpretive symmetry, and compressive drift by reinforcing distinctions that risk being collapsed (Ganguli et al., 2023)[12]. When grounding weakens, context becomes more susceptible to drift away from the originating purpose, even if the surface content remains relevant (Shelton, 2025)[2].

7.2 Structural Protocols and Cohesion Maintenance

Structural protocols define the organizational form of reasoning. They counteract cohesion drift by preserving relationships between context components, recursive drift by imposing interpretable boundaries between reasoning layers, and compressive drift by segmenting dense or complex material (Wei et al., 2022)[4]; (Zhou et al., 2023)[7]. When structural forces erode, multi-step reasoning collapses into blended analysis or ambiguous interpretation (Bommasani et al., 2021)[14].

7.3 Drift-Management Protocols and Directional Balance

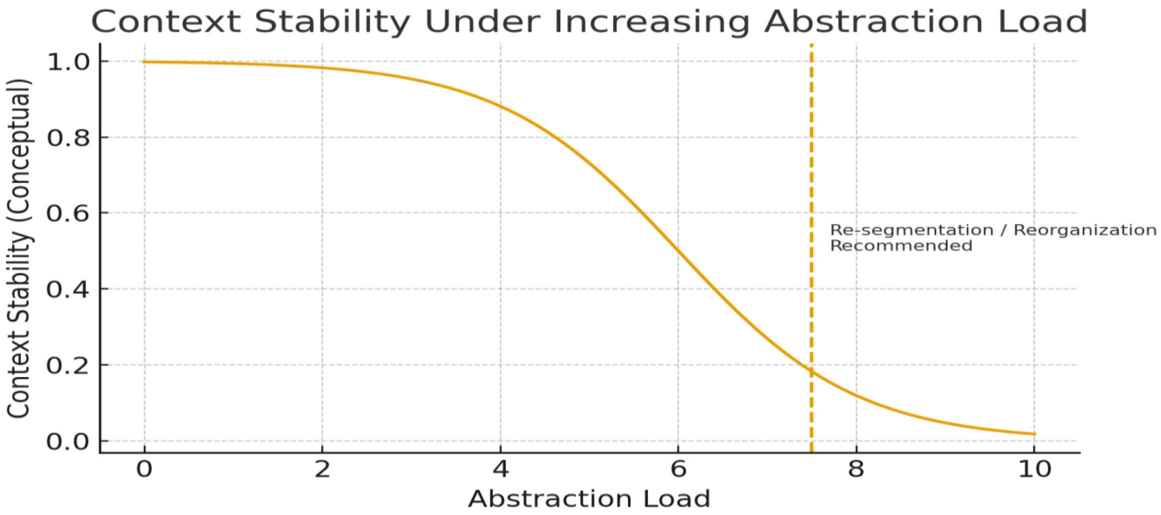
Drift-management protocols provide the conceptual lens through which deviations in interpretive trajectory can be recognized or constrained. They counteract directional drift by identifying imbalances in perspective or emphasis, abstraction drift by maintaining consistency between detail-level and high-level reasoning, and recursive drift by preventing runaway abstraction during iterative reinterpretation (Ganguli et al., 2023)[12].

7.4 Reflection Protocols and Recursive Stability

Reflection protocols govern how prior reasoning is evaluated or refined. They counteract recursive drift by maintaining continuity between iterations, abstraction drift by ensuring reinterpretation does not escalate into thematic generalization, and cohesion drift by preserving linkages across revisions (Wei et al., 2022)[4]. Reflection is structurally necessary, but without constraints it can amplify drift. Reflection protocols define conceptual boundaries for safe reinterpretation (Zhou et al., 2023)[7].

7.5 Context-Integrity Protocols and Stability Conditions

Context-integrity protocols define the conditions under which growing contexts remain interpretable. They counteract compressive drift by defining thresholds for re-segmentation or reorganization, cohesion drift by preserving structural coherence under high context density, and abstraction drift by regulating how much conceptual material can accumulate without distortion (Anthropic, 2023)[10]; (OpenAI, 2023)[11].



7.6 Interactions as Structural Dynamics

The interactions between protocol forces and drift pressures create predictable structural dynamics within context-driven systems: when grounding is weak → directional and abstraction drift increase; when structure is weak → cohesion and recursive drift emerge; when reflection is unconstrained → recursive drift accelerates; when context integrity is exceeded → compressiveness and collapse patterns appear (Ganguli et al., 2023)[12]; (Bowman, 2023)[13]. Rather than treating drift as error, Protocol Governance frames drift as the structural consequence of unregulated context evolution (Shelton, 2024)[1]. Protocols provide the conceptual architecture needed to understand, analyze, and eventually stabilize these dynamics in future work (Bommasani et al., 2021)[14].

8. Implications for Research and System Design

The integration of protocol taxonomy and drift mechanics offers more than a theoretical lens; it establishes a foundation for systematic inquiry into context-driven AI systems (Bommasani et al., 2021)[14]. These implications extend across research methodology, system design, interpretability, and future theoretical development. While Protocol

Governance is intentionally non-mechanistic, it informs conceptual tools necessary for empirical and architectural progress (Kaplan et al., 2020)[15].

8.1 Implications for Research Methodology

Protocol Governance provides a structured vocabulary for analyzing context behavior, enabling researchers to identify drift patterns with greater precision, classify reasoning instabilities using a unified taxonomy, evaluate multi-turn interactions through structural criteria, differentiate between content-related failures and structure-related failures, and develop hypotheses using well-defined conceptual constructs (Bowman, 2023)[13]. This structural framing supports the emergence of context-centric analysis, where context itself becomes the object of study rather than a secondary artifact of prompting (Liu et al., 2023)[6].

8.2 Implications for System Design

While this framework does not prescribe architectural changes, it highlights design considerations for systems involving prompting, retrieval, agentic orchestration, or memory components. Grounding forces suggest the importance of mechanisms that maintain persistent connection to task constraints; structural forces highlight the value of organizing multi-step reasoning into interpretable, segmented forms; drift-management insights help designers recognize where iterative systems require balancing or recalibration; reflection structures provide conceptual clarity for systems that reinterpret prior outputs; integrity conditions indicate thresholds for context size, abstraction load, or re-segmentation (Lewis et al., 2020)[8]; (Wang et al., 2023)[5]; (Anthropic, 2023)[10].

8.3 Implications for Cross-Model Analysis

Because Protocol Governance is architecture-agnostic, it provides universal axes along which models can be compared: susceptibility to specific drift modes; stability of segmentation boundaries; resilience to abstraction load; consistency of grounding under multi-turn tasks; and integrity maintenance in long interactions (OpenAI, 2023)[11]; (Bommasani et al., 2021)[14]. This creates a shared foundation for evaluating models based on structural behavior rather than task accuracy alone (Kaplan et al., 2020)[15].

8.4 Implications for Interpretability

Protocol Governance supports a form of interpretability grounded in context evolution, rather than internal mechanisms. It offers a conceptual explanation for why reasoning diverges, a descriptive model for how drift accumulates, a structural lens for identifying instability, and a vocabulary that maps directly to observable behavior (Bowman, 2023)[13]. Interpretability, in this framing, comes from understanding the shape of context, not the opacity of model internals (Bommasani et al., 2021)[14].

8.5 Implications for Future Theory

Finally, this framework opens the door to a broader discipline: a structural theory of context, analogous to theoretical frameworks in systems theory, linguistics, or cognitive science. Such a discipline could explore formal definitions of protocol categories, quantitative measurements of drift, stability boundaries in multi-turn contexts, conceptual models of context evolution, and structural constraints that predict collapse patterns (Kaplan et al., 2020)[15].

9. Toward a Structural Theory of Context

The preceding sections have defined the conceptual components required to understand context-driven AI systems: protocol categories, drift mechanics, and the structural interactions between them. Together, these elements suggest the emergence of a broader theoretical discipline—a structural theory of context—in which context is treated as a formal object of study with definable properties, constraints, and behaviors (Bommasani et al., 2021)[14].

This theoretical shift parallels developments in other fields. Linguistics treats language as a system governed by structural rules rather than merely a set of utterances. Systems theory studies feedback, stability, and organization without requiring access to internal mechanisms. Complexity science identifies emergent patterns that arise from interactions of simple components. In each case, structure is the central analytic unit (Kaplan et al., 2020)[15].

Similarly, a structural theory of context would treat context not as incidental text that happens to precede a model's output, but as a dynamic structure whose evolution is shaped by identifiable forces—forces that can be described, analyzed, and eventually formalized (Bowman, 2023)[13].

Several foundational principles begin to emerge:

Context has properties: integrity, segmentation, relevance boundaries, abstraction load, and susceptibility to drift—observable features, not incidental artifacts (Anthropic, 2023)[10].

Context evolves predictably: multi-turn or multi-layer interactions reveal consistent patterns of drift, collapse, and reorganization, forming theoretical principles similar to stability conditions (Wei et al., 2022)[4].

Context is shaped by structural forces: protocols as conceptual forces govern how context maintains coherence (Shelton, 2024)[1].

Context behavior is architecture-agnostic: principles apply across model families, scales, and design philosophies; drift patterns emerge from interaction dynamics, not internal states (OpenAI, 2023)[11].

Context is analyzable without anthropomorphism: a structural theory treats context as externally visible, text-based artifacts (Bommasani et al., 2021)[14].

Context stability is central to future AI design: as systems rely more on multi-step reasoning, tool use, and long-context behavior, understanding context structure becomes indispensable (Kaplan et al., 2020)[15].

Taken together, these principles point toward a future research landscape in which context is studied with rigor similar to other complex systems. Protocol Governance provides the conceptual foundation for such a field. It does not claim to be complete or final; rather, it establishes the initial vocabulary, categories, and structural observations required for formal advancement (Shelton, 2025)[2].

A structural theory of context will enable systematic comparison of context behaviors, investigation of stability boundaries, formal definitions of drift trajectories, development of theoretical models of context evolution, deeper understanding of failure modes in multi-turn reasoning, and ultimately positions context as a structured medium with rules and dynamics of its own (Bommasani et al., 2021)[14].

10. Limitations and Scope

Protocol Governance is introduced in this work as a conceptual framework, not as an empirical claim or architectural proposal. Its purpose is to define the structural vocabulary needed to analyze context behavior—not to prescribe mechanisms for how models should be built or how systems should enforce stability (Shelton, 2024)[1].

10.1 Conceptual, Not Mechanistic

This paper does not propose new model architectures, training methods, or inference-time algorithms. Protocols are conceptual constructs, not internal rules executed by models. They describe structural expectations external observers can use to analyze context behavior, not processes the model performs internally (Bommasani et al., 2021)[14].

10.2 No Claims About Cognition or Agency

The framework intentionally avoids any implications of internal goals, monitoring, self-regulation, reflective cognition, or intentional drift correction. Drift, collapse, and structural forces belong to the context, not the model. This focus preserves conceptual neutrality and aligns with non-anthropomorphic interpretive standards (Bowman, 2023)[13].

10.3 Not a Replacement for Prompting, RAG, or ACE

Protocol Governance does not supersede or replace existing methods such as prompt engineering, retrieval-augmented generation (RAG), memory systems, agentic workflows, or ACE. Instead, it provides the structural language required to understand why these methods succeed or fail under specific conditions (Liu et al., 2023)[6]; (Lewis et al., 2020)[8]; (Author et al., 2025)[3].

10.4 No Quantitative Claims (Yet)

This work does not present metrics, empirical evaluations, statistical measurements, or benchmarking results. Such work is future-facing. The framework is intentionally qualitative at this stage (Kaplan et al., 2020)[15].

10.5 Context as the Only Analytic Object

Protocol Governance deliberately focuses only on context, its structure, its evolution, and its drift patterns. It does not attempt to make claims about model internals, gradient behavior, neural activations, hidden states, or scaling laws. This limitation is intentional to preserve cross-architecture compatibility (OpenAI, 2023)[11].

10.6 No Mandated Implementation

While future work may draw inspiration from these concepts, this paper does not propose enforcement mechanisms, guardrails, system modifications, inference pipelines, or algorithmic interventions. Any such developments would require separate empirical validation and safety analysis (Bommasani et al., 2021)[14].

10.7 Boundaries of Interpretive Use

The framework is not intended as a diagnostic tool for evaluating model cognition, evidence for or against agentic behavior, a claim about mental states, or a justification for narratives of emergent intelligence. It is a structural framework for analyzing context behavior—nothing more, nothing less (Bowman, 2023)[13].

10.8 Summary of Scope

Protocol Governance is best understood as a conceptual foundation, a descriptive framework, a vocabulary for analysis, and a precursor to empirical study. Its purpose is to establish the theoretical architecture necessary for future research into context stability, drift dynamics, and structural behavior across systems (Shelton, 2024)[1]; (Shelton, 2025)[2].

11. Future Directions

While Protocol Governance is presented here as a conceptual foundation, it opens a variety of research paths across theoretical, empirical, and comparative domains. These directions build on the structural vocabulary and analytical framing established in earlier sections (Bommasani et al., 2021)[14].

11.1 Formalization of Protocol Categories

An immediate direction is the formal definition of protocol categories, including precise criteria for grounding, structure, drift management, reflection, and integrity; formal relationships between protocol types; and definitional boundaries and equivalence classes (Shelton, 2024)[1]. Such work could support mathematical or computational formalizations of context behavior without departing from the non-mechanistic nature of the framework (Kaplan et al., 2020)[15].

11.2 Metrics for Drift and Context Stability

The drift modes outlined earlier invite the creation of measurable, quantifiable indicators of context stability: drift trajectories, collapse signatures, interpretive stability metrics, and multi-turn reasoning coherence indices. These metrics would enable cross-model comparison and form the basis for systematic evaluation (Bowman, 2023)[13]; (OpenAI, 2023)[11].

11.3 Protocol-Inspired Evaluation Frameworks

Evaluation frameworks informed by Protocol Governance could analyze grounding fidelity across turns, structural stability of reasoning chains, resilience to abstraction load, and robustness against drift under recursive refinement (Wei et al., 2022)[4]; (Zhou et al., 2023)[7].

11.4 Comparative Analysis Across Model Families

Since the framework is architecture-agnostic, future work can explore differences across transformers, mixture-of-experts models, smaller finetuned architectures, agentic systems, retrieval-augmented systems, and memory-augmented systems. Comparative research could identify characteristic drift patterns unique to specific design families (Bommasani et al., 2021)[14]; (Lewis et al., 2020)[8].

11.5 Integration With Existing Approaches

Protocol Governance complements, rather than replaces, existing methods. Future work may explore how structural concepts interact with ACE, RAG pipelines, memory frameworks, instruction hierarchies, and orchestrated agent loops (Author et al., 2025)[3]; (Lewis et al., 2020)[8]; (Wang et al., 2023)[5].

11.6 Structural Context Simulations

Researchers could build simulations modeling context evolution under varying conditions of drift pressure, abstraction load, segmentation changes, and multi-turn iteration. Such simulations would provide a controlled environment for observing structural dynamics (OpenAI, 2023)[11].

11.7 Educational and Pedagogical Applications

The clarity of the protocol taxonomy and drift mechanics provides foundations for graduate seminars, interpretability workshops, AI safety courses, and interdisciplinary teaching. Educators can use the framework to teach context behavior without requiring architectural detail (Bommasani et al., 2021)[14].

11.8 Toward a Structural Theory of Context

Ultimately, Protocol Governance may evolve into a broader discipline dedicated to formal context analysis, structural modeling, drift dynamics, context stability boundaries, and cross-model generalization (Kaplan et al., 2020)[15].

12. Conclusion

Context-driven systems have become central to modern AI, yet their behavior remains difficult to analyze without a structural framework. This work introduced Protocol Governance as a conceptual architecture for understanding how context forms, evolves, and stabilizes during extended reasoning (Shelton, 2024)[1]; (Shelton, 2025)[2]. By distinguishing context content from structural forces, and by articulating both a protocol taxonomy and recognizable drift patterns, we have outlined the foundations for a structural theory of context (Bommasani et al., 2021)[14].

Protocols—grounding, structural, drift-management, reflection, and context-integrity—serve as conceptual elements that govern context interpretation and coherence. Drift mechanics reveal predictable modes by which context degrades in their absence. Together, these perspectives create a unified vocabulary for analyzing context behavior across systems, tasks, and architectures, and provide a starting point for future theoretical, empirical, and comparative research into context dynamics (Kaplan et al., 2020)[15].

This framework is intentionally non-mechanistic. It does not prescribe implementation details or propose architectural interventions. Instead, it offers the clarity needed to treat context as a formal object of study—one whose stability, coherence, and interpretive boundaries can be described conceptually without anthropomorphism or appeals to internal cognitive processes (Bowman, 2023)[13].

As research advances, Protocol Governance may serve as the foundation for a broader discipline centered on the structural dynamics of context. The ideas introduced here—protocol patterns, drift modes, and structural interactions—establish the scaffold upon which future work can construct metrics, formal models, evaluation frameworks, and cross-model comparisons (OpenAI, 2023)[11]. By defining the conceptual architecture underlying context evolution, this monograph provides the groundwork for a more rigorous understanding of modern AI behavior.

Appendix A — Glossary of Structural Terms

This glossary provides formal definitions for the core concepts used throughout this monograph. The terms are organized to support clarity, consistency, and ease of reference for future research, teaching, and structural analysis (Bommasani et al., 2021)[14].

Context-related terms

Context: the evolving set of information, instructions, and prior model outputs that influence a model's responses during an interaction. Context is treated as a structural object, not a cognitive state (Bowman, 2023)[13].

Context evolution: the process by which context changes over time through accumulation, transformation, or reorganization across multi-step interactions (Wei et al., 2022)[4].

Context integrity: the condition in which context remains coherent, interpretable, and aligned with task goals across extended reasoning (Anthropic, 2023)[10].

Context collapse: a degradation state in which context loses structure, becomes overloaded, or diverges from task alignment, leading to instability in reasoning (OpenAI, 2023)[11].

Multi-turn interaction: extended exchanges in which context evolves across multiple reasoning steps, often amplifying structural pressures such as drift, abstraction layering, or relevance decay (Zhou et al., 2023)[7].

Task alignment: the degree to which context maintains connection to the original user goals, constraints, and priorities as it evolves (Shelton, 2025)[2].

Drift terms

Drift: a structural deviation in context interpretation or reasoning trajectory that emerges during context evolution (Ganguli et al., 2023)[12].

Compressive drift: loss of distinctions or specificity as information density increases, causing the context to collapse toward generalized representations (Anthropic, 2023)[10].

Abstraction drift: a shift from concrete, detail-rich reasoning toward higher-level thematic or generalized analysis (Bowman, 2023)[13].

Directional drift: a bias in which reasoning increasingly favors one interpretive direction, perspective, or emphasis (Ganguli et al., 2023)[12].

Recursive drift: a compounding drift mode where reinterpretations of prior outputs gradually amplify abstraction, narrowing, or distortion (Wei et al., 2022)[4].

Cohesion drift: degradation of the relationships between contextual components even when the components themselves remain individually relevant (Bommasani et al., 2021)[14].

Abstraction load: structural pressure placed on context when high-level conceptual material accumulates, increasing susceptibility to drift and loss of interpretive stability (OpenAI, 2023)[11].

Protocol categories

Protocols: conceptual constructs that describe structural expectations governing context interpretation, transformation, and stability. Protocols are not rules executed by models, but frameworks for analyzing context behavior (Shelton, 2024)[1].

Grounding protocols: define how context remains tied to original user intent and task boundaries (Shelton, 2025)[2].

Structural protocols: define the organizational form of reasoning, including segmentation, stepwise progression, and interpretive boundaries (Wei et al., 2022)[4].

Drift-management protocols: describe how deviations in interpretive trajectory are conceptually identified and constrained (Ganguli et al., 2023)[12].

Reflection protocols: define structural conditions under which prior reasoning can be reconsidered or refined without inducing recursive drift (Zhou et al., 2023)[7].

Context-integrity protocols: define conditions required for context to remain coherent and interpretable as it accumulates complexity (Anthropic, 2023)[10].

Structural analysis terms

Interpretive boundary: a conceptual division that separates distinct reasoning steps, abstraction levels, or task components (Wei et al., 2022)[4].

Stability condition: a structural requirement that preserves coherence or prevents drift during context evolution (Kaplan et al., 2020)[15].

Drift trajectory: an abstract path through which drift progresses over time, driven by structural pressures (Ganguli et al., 2023)[12].

Segmentation: conceptual partitioning of context into organized units such as steps, phases, or structured components (Zhou et al., 2023)[7].

Relevance boundary: the constraint defining what information remains task-relevant during multi-step interactions (Liu et al., 2023)[6].

Meta-framework terms

Protocol Governance: the structural framework introduced in this monograph for analyzing how context is governed through protocols and drift dynamics (Shelton, 2024)[1].

Structural theory of context: a prospective academic discipline centered on studying context as a formal, analyzable system with definable structural properties (Bommasani et al., 2021)[14].

Structural forces: conceptual forces such as grounding, segmentation, or coherence that shape how context behaves, without implying mechanism or cognition (Bowman, 2023)[13].

Appendix B — Comparative Notes on ACE and Protocol Governance

This appendix clarifies the relationship between Stanford's Agentic Context Engineering (ACE) and the conceptual framework developed in this monograph. While both explore the importance of context in modern AI systems, they operate at different layers with distinct aims and theoretical commitments (Author et al., 2025)[3].

Scope and intent

ACE: examines how agentic systems construct, manipulate, and evolve context as part of iterative workflows—action-oriented task completion, adaptive context shaping, agentic loops, multi-step orchestration (Wang et al., 2023)[5].

Protocol Governance: defines conceptual structures that govern context behavior itself, regardless of agentic design—stability, drift dynamics, structural constraints, interpretive boundaries (Shelton, 2024)[1].

Distinction: ACE optimizes context engineering; Protocol Governance describes context governance (Author et al., 2025)[3].

Conceptual layer

ACE: operates at the operational layer—how context is built, how agents modify it, how iterative steps interact with it (Wang et al., 2023)[5].

Protocol Governance: operates at the structural layer—how context maintains coherence, how structural forces guide interpretation, how drift and collapse emerge, how context evolves as a system (Bommasani et al., 2021)[14].

Distinction: ACE explores what context can do; Protocol Governance explains how context behaves (Author et al., 2025)[3].

Relationship to drift and stability

ACE: encounters drift as a practical obstacle to agentic workflows (Wang et al., 2023)[5].

Protocol Governance: frames drift as a structural phenomenon caused by unregulated context evolution (Ganguli et al., 2023)[12].

Distinction: ACE mitigates drift procedurally; Protocol Governance analyzes drift conceptually (Shelton, 2024)[1].

Underlying assumptions

ACE: assumes agentic structure, multi-step planning, self-referential loops, adaptive modification of context (Wang et al., 2023)[5].

Protocol Governance: assumes no agency, no internal state, no cognitive monitoring, and context as the only analytic object (Bowman, 2023)[13].

Distinction: ACE is agentic; Protocol Governance is non-agentic and architecture-neutral (Bommasani et al., 2021)[14].

Complementarity The two approaches are synergistic, not competitive. ACE benefits from protocol insights describing drift and stability; Protocol Governance benefits from ACE's empirical demonstrations that context evolution matters (Author et al., 2025)[3]; (Wang et al., 2023)[5].

6. Summary Table

Aspect	ACE	Protocol Governance
Domain	Agentic workflows	Structural analysis
Focus	Context manipulation	Context behavior
Assumptions	Agentic loops	Statelessness
Handles Drift	Practically	Conceptually
Contribution	Demonstrates utility	Defines architecture

Appendix C — Illustrative Structural Scenarios

This appendix provides abstract, non-architectural scenarios illustrating how protocol categories and drift patterns manifest in context-driven interactions. These examples clarify structural behavior of context under different conditions (Bowman, 2023)[13]; (Wei et al., 2022)[4].

SCENARIO 1: GROUNDING PROTOCOL — MAINTAINING TASK FOCUS

Setup

The user requests "Summarize only the financial risks of the proposal." After multiple turns, the context contains references to staffing, scheduling, logistics, and technical dependencies.

Observed Pattern

Summaries drift toward general risk categories rather than strictly financial risks.

Structural Interpretation

Grounding forces weakened as unrelated material accumulated, enabling abstraction drift (Ganguli et al., 2023)[12].

SCENARIO 2: STRUCTURAL PROTOCOL — PRESERVING MULTI-STEP FORMAT

Setup

A user provides a 3-step reasoning structure—identify constraints; analyze impacts; propose mitigations. After iterations, outputs merge steps or omit initial constraint identification.

Observed Pattern

Cohesion drift as segmentation boundaries weaken.

Structural Interpretation

Structural protocols conceptually maintain stepwise clarity; their absence leads to blended reasoning (Wei et al., 2022)[4]; (Zhou et al., 2023)[7].

SCENARIO 3: DRIFT-MANAGEMENT PROTOCOL — BALANCING INTERPRETIVE TRAJECTORY

Setup

"List benefits and drawbacks of Proposal A." Negative phrasing accumulates in the context via clarifications or concerns.

Observed Pattern

Analysis shifts toward drawbacks disproportionately.

Structural Interpretation

Directional drift emerges due to interpretive imbalance; drift-management protocols conceptually define where rebalancing is needed (Ganguli et al., 2023)[12].

SCENARIO 4: REFLECTION PROTOCOL — PREVENTING RECURSIVE ABSTRACTION

Setup

Summarize → refine summary → refine refinement → repeat.

Observed Pattern

Highly abstract output loosely related to original content.

Structural Interpretation

Recursive drift causes compounding abstraction and loss of detail; reflection protocols define continuity checkpoints (Wei et al., 2022)[4].

SCENARIO 5: CONTEXT-INTEGRITY PROTOCOL — FRAGMENTATION UNDER LOAD

Setup
Decision-making sequence accumulates constraints, assumptions, long-term goals, new data, contradictions, and revised requirements.

Observed Pattern
Cohesion drift arises as relational structure degrades under abstraction load.

Structural Interpretation
Integrity protocols indicate when re-segmentation or reorganization becomes necessary (Anthropic, 2023)[10]; (OpenAI, 2023)[11].

Appendix D — Neutrality, Non-Agency, and Structural Framing

This appendix establishes the philosophical and methodological boundaries for interpreting Protocol Governance correctly. Structural terms—drift, grounding, reflection, integrity—do not imply cognition, introspection, or agency; they describe context, not models (Bowman, 2023)[13].

Non-agency is foundational Protocol Governance does not attribute agency, intent, goals, or self-regulation to AI systems. Protocols provide a conceptual scaffold for external observers to analyze context evolution; they are descriptive abstractions, not internal mechanisms (Bommasani et al., 2021)[14].

Statelessness and non-cognition The framework treats LLMs as stateless token predictors. "Evolution" refers to text accumulation, not memory; "reflection" refers to reinterpretation within text, not introspection; "stability" refers to structural coherence, not cognitive consistency; "interpretation" is an external analytic term (OpenAI, 2023)[11].

Structural forces are conceptual, not mechanistic Terms such as "pressure," "drift," "force," or "integrity" describe structural patterns in external textual artifacts, not internal computational states. They parallel constraints in linguistics, dynamics in systems theory, and stability in control theory without implying emergent goals or self-regulation (Kaplan et al., 2020)[15].

Context behavior is distinct from model behavior The model predicts tokens; the context behaves structurally. Drift, collapse, and integrity are properties of how context transforms, not of the underlying model (Bowman, 2023)[13].

No claims of intentional correction or monitoring Although the framework describes drift-management and reflection protocols, it does not suggest that models detect deviation, correct errors, resolve contradictions, or maintain internal consistency. These protocols exist for analytical purposes only (Ganguli et al., 2023)[12].

Alignment with ethical and safety standards Protocol Governance avoids anthropomorphism, hidden-state inference, claims of emergent cognition, agentic interpretation, and speculative narratives about model intent. Its neutrality suits academic, safety-oriented, and interpretability-focused research (Bommasani et al., 2021)[14].

Conceptual neutrality across architectures The framework assumes nothing about specific model types, datasets, training regimes, scaling, infrastructure, or agentic design. It applies universally because it analyzes context, not models (OpenAI, 2023)[11].

Reading and applying the framework safely Use this framework as a vocabulary for context behavior, a lens for drift and collapse, a structural blueprint for future theory, and a conceptual foundation for empirical work—without implying how AI "thinks" (Kaplan et al., 2020)[15].

References

Core Theoretical Foundations [1] Shelton, C. (2024). Cross-Architecture Constructive Interference Model (CACIM). OSF Preprints. https://doi.org/10.17605/OSF.IO/Q37ZR [2] Shelton, C. (2025). Compressed Structured Interaction Protocol (CSIP): A Reduced-Overhead Metacognitive Framework for Small and Efficiency-Oriented LLMs. OSF Preprints. https://doi.org/10.17605/OSF.IO/F74X6

Context Engineering and Agentic Systems [3] [Stanford ACE paper - awaiting publication details]. Agentic Context Engineering. Stanford University. (2025). [4] Wei, J., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35, 24824-24837. [5] Wang, L., et al. (2023). Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. Proceedings of ACL 2023. [7] Zhou, D., et al. (2023). Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. ICLR 2023.

Prompt Engineering and Context Management [6] Liu, P., et al. (2023). Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. ACM Computing Surveys, 55(9), 1-35.

Memory and Retrieval Systems [8] Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS, 33, 9459-9474. [9] Borgeaud, S., et al. (2022). Improving language models by retrieving from trillions of tokens. ICML, 2206-2240.

Context Window and Long-Form Reasoning [10] Anthropic. (2023). Claude's Constitution: Training a Harmless AI Assistant. Anthropic Technical Report. [11] OpenAI. (2023). GPT-4 Technical Report. arXiv:2303.08774.

Drift and Stability in AI Systems [12] Ganguli, D., et al. (2023). The Capacity for Moral Self-Correction in Large Language Models. arXiv:2302.07459. [13] Bowman, S. R. (2023). Eight Things to Know about Large Language Models. arXiv:2304.00612.

Theoretical Frameworks for AI Behavior [14] Bommasani, R., et al. (2021). On the Opportunities and Risks of Foundation Models. Stanford CRFM Technical Report. [15] Kaplan, J., et al. (2020). Scaling Laws for Neural Language Models. arXiv:2001.08361.