Hop Scaling: Multi-Agent Chain Degradation Saturates

Jeremy McEntire[1]

March 2026

Abstract

Paper XXIII established that the context fence + domain prime handoff unit achieves CE = 0.503 bilaterally. This paper tests whether that unit scales to N-hop sequential chains. Four experiments on Qwen 2.5-7B measure chain degradation across 1–10 hops, re-priming interventions, shared vocabulary substrates, and cross-domain handoffs.

The headline finding: chain degradation *saturates*. CE rises from 0.503 at hop 1 to ∼0.60 by hop 5, then plateaus through hop 10. Neither linear ($R^2 = 0.50$) nor exponential ($R^2 = 0.48$) models fit well because the actual shape is logarithmic — fast initial rise, then flat. The protocol scales: per-hop degradation does not compound.

Three secondary findings constrain swarm architecture design. First, fencing at every hop versus only the terminal hop produces identical results (0.598 vs. 0.597 at 10 hops), confirming that intermediate fences are pure overhead — the fence prepares the worker, not the relay chain. Second, pre-loading shared vocabulary into all chain agents *hurts* at every chain length (+0.04 to +0.07 CE), replicating Paper XX's saturation effect at the chain level. Third, cross-domain handoffs (medical→legal, code→science) outperform within-domain chains (CE 0.56–0.63 vs. 0.72), because the domain switch forces a genuine mode reset that within-domain chains never trigger.

1 Motivation

Papers XVI–XXIII established a bilateral coordination protocol: context fence (mode-switching boundary) followed by domain prime (15–50 tokens of domain content). The protocol achieves CE = 0.503, closing 163% of the neutral-to-expert gap. Paper XXIII showed the two stages are separable and compose independently ($\rho = 0.858$).

The bilateral result leaves the scaling question open. In any multi-agent system — Pact's component trees, Claude Code's subagent chains, production LLM pipelines — information passes through multiple handoffs. Does the coordination quality degrade linearly with hop count (manageable), exponentially (catastrophic), or does it saturate (benign)?

The answer determines whether multi-agent systems need shallow hierarchies for information-theoretic reasons or whether depth is constrained only by the $O(n^2)$ coordinator complexity that organizational theory identifies as the binding limit.

[1]Correspondence: jmc@cageandmirror.com

2 Method

2.1 Chain construction

Each chain of length N works as follows. The first agent receives a domain-primed probe and generates a 48-token response. Each subsequent agent receives the prior agent's response as context and generates its own 48-token continuation. The final agent's processing state is measured against the expert's continuation of the original probe using sender continuation perplexity (Paper XVIII's KV-cache method).

Three fence modes are tested:

- **Fence every hop**: context fence + domain prime prepended at each handoff.
- **Fence final only**: no fences at intermediate hops; fence + prime only at the terminal agent.
- **No fence**: bare context relay throughout.

Chain lengths: $N \in \{1, 2, 3, 5, 7, 10\}$. Model: Qwen 2.5-7B. 40 probes (10 per domain: medical, legal, code, science). Expert continuations: 64 tokens generated from domain-primed context.

2.2 Re-priming intervention (Experiment 2)

A 7-hop chain with fences at every hop, with a full re-priming intervention (reset to expert context rather than continuing the chain) injected at hop 3 or hop 5. Tests whether mid-chain re-priming can reduce the plateau level.

2.3 Shared vocabulary substrate (Experiment 3)

All chain agents receive a domain vocabulary preamble before the chain begins. Tests the hypothesis that a shared coordination substrate reduces per-hop degradation by maintaining vocabulary alignment across the chain.

2.4 Cross-domain handoffs (Experiment 4)

Three-hop chains where the intermediate agent operates in a different domain than the source and terminal agents. Domain pairs: medical→legal, code→science, legal→code. Control: within-domain medical→medical.

3 Results

3.1 Experiment 1: Chain degradation saturates

Table 1: CE at each hop count by fence mode. All values are means across 40 probes (4 domains × 10 probes). Baseline (hop 0, fence+prime) = 0.503.

	Hop count					
Fence mode	1	2	3	5	7	10
Every hop	0.503	0.560	0.590	0.607	0.602	0.598
Final only	0.503	0.557	0.583	0.596	0.596	0.597
No fence	0.596	0.651	0.679	0.708	0.708	0.708

Three observations.

Saturation, not compounding. All three fence modes plateau by hop 5. The fenced conditions asymptote at ∼0.60; the unfenced condition at ∼0.71. Neither linear ($R^2 = 0.50$, slope = 0.008) nor exponential ($R^2 = 0.48$, rate = 0.015) models fit well. The actual shape is logarithmic: a step increase from hop 1 to hop 2 (+0.057), diminishing increments through hop 5 (+0.017), and no further change through hop 10 (−0.009 for *fence_every_hop*; +0.001 for *fence_final_only*). The hop 7 and hop 10 values for the fenced conditions are *lower* than hop 5, consistent with regression to a stable equilibrium rather than monotonic growth.

Intermediate fences are overhead. At 10 hops, *fence_every_hop* (0.598) and *fence_final_only* (0.597) differ by 0.001 nats. The intermediate fences at hops 2–9 contribute nothing measurable. This directly confirms Paper XXIII's mode-switching interpretation: the fence's function is to install a processing boundary at the terminal agent — the one that actually does the domain-specific work. Relay agents are passing tokens, not processing domain content; fencing them is applying the wrong tool to the wrong problem.

The fence reduces the asymptote, not the slope. Both fenced conditions have slopes of ∼0.008; the unfenced has slope ∼0.011. The slopes are comparable. What the fence changes is the intercept and the plateau level: 0.60 vs. 0.71, a 15% reduction in the equilibrium CE. The fence does not prevent degradation — it lowers the floor.

3.2 Experiment 2: Re-priming has a timing dependency

Re-priming at hop 3 slightly *worsens* performance (+0.007). At hop 3, the chain has not yet reached the plateau; the re-priming intervention disrupts a trajectory that was still converging. Re-priming at hop 5 *helps* (−0.043), resetting the chain from its plateau level and giving the final two hops a fresh start.

Table 2: Re-priming intervention in a 7-hop chain with fences at every hop.

Condition	CE
No re-prime	0.602
Re-prime at hop 3	0.609
Re-prime at hop 5	0.560

The improvement from hop-5 re-priming (CE = 0.560) is modest — it reduces the 7-hop chain to approximately the 2-hop level. Re-priming is not free: it replaces the chain's accumulated context with the expert's original context, discarding whatever domain-relevant content the chain generated. The benefit exists only when the chain's accumulated noise exceeds the information value of its accumulated content, which occurs at or after the saturation point.

3.3 Experiment 3: Shared vocabulary hurts

Table 3: Effect of shared vocabulary substrate. All chains use *fence_every_hop* mode.

Hops	Without vocab	With vocab	Δ
1	0.503	0.570	+0.067
3	0.590	0.630	+0.040
5	0.607	0.653	+0.046
7	0.602	0.646	+0.044

Shared vocabulary consistently *increases* CE by +0.04 to +0.07 across all chain lengths. The effect is largest at hop 1 (+0.067) and stabilizes around +0.045 at longer chains.

This replicates Paper XX's saturation finding at the chain level. The domain prime at each hop already activates the correct processing mode. The vocabulary preamble adds tokens beyond the optimal window, pushing each agent past the ∼150-token saturation point where additional context degrades rather than helps. The shared vocabulary is not aligning the chain — it is adding a consistent noise floor at each hop.

The implication for multi-agent architecture: do not pre-load agents with shared context. Let each agent prime independently from its own handoff brief. The contract (in Pact's architecture) serves as the shared vocabulary naturally — each agent reads it fresh, generating its own internal representation rather than receiving a pre-digested one.

Table 4: Three-hop chains with cross-domain intermediate agents.

Chain path	CE
Within-domain (med→med→med)	0.718
Medical → Legal → Medical	0.630
Code → Science → Code	0.559
Legal → Code → Legal	0.565

3.4 Experiment 4: Cross-domain handoffs outperform within-domain

Cross-domain chains outperform within-domain chains by 0.09–0.16 CE. The within-domain chain (0.718) is markedly worse than any cross-domain condition.

The mechanism: within-domain chains accumulate domain-*specific* interference. Each relay agent generates domain-matched content that shifts the trajectory further into the domain's processing mode — but along a path the expert did not take. The accumulated content is domain-relevant but trajectory-divergent.

Cross-domain handoffs break this accumulation. The domain switch at the intermediate hop forces a genuine processing mode reset — not because a fence instruction was present, but because the domain content itself is discontinuous. The intermediate agent's legal or science content is semantically orthogonal to the source domain's medical or code content. This orthogonal interruption acts as a natural fence, clearing domain-specific accumulation before the terminal agent re-primes into the original domain.

This is Paper XXIII's mode-switching mechanism operating naturally. The explicit context fence achieves the same effect artificially. The cross-domain result shows that mode discontinuity is the active ingredient, whether introduced by explicit instruction or by content properties.

4 The Saturation Mechanism

Why does chain degradation saturate rather than compound? The data suggests an equilibrium interpretation.

Each relay agent generates text from a domain-primed state. This text is domain-relevant but trajectory-divergent: it carries domain vocabulary and reasoning patterns but follows the relay agent's own processing trajectory rather than the expert's. At the next hop, the receiving agent processes this text.

The first hop introduces the largest trajectory divergence, because the relay agent's response is maximally different from the expert's original context. By hop 3–5, the relay agents' responses have converged to a *chain equilibrium* — a stable distribution of domain-relevant content that carries domain identity but no further expert-specific trajectory information. Additional hops produce responses drawn from the same equilibrium distribution. The CE stabilizes because the input distribution has stabilized.

The fence controls the *level* of this equilibrium. With fences, each relay agent starts from a cleaned state, producing responses that are less domain-entangled. Without fences, the relay agents accumulate domain interference, producing a higher-CE equilibrium. But both equilibria are stable.

This has a direct architectural implication: **chain depth is not a binding constraint for multi-agent coordination**. A 10-agent pipeline performs the same as a 5-agent pipeline. The binding constraints on swarm architecture are the $O(n^2)$ coordinator complexity (the Graicunas limit) and the per-hop time cost, not information-theoretic degradation.

5 Implications for Swarm Design

5.1 What the data says to build

Fence the worker, not the relay. Only the terminal agent — the one performing domain-specific reasoning — needs the context fence + domain prime. Intermediate relay agents can pass context without fences (0.001 nats difference at 10 hops). This reduces coordination overhead for deep pipelines.

Don't pre-load shared context. Let each agent prime from its own handoff brief. Shared vocabulary substrates add consistent noise (+0.04–0.07 CE at all chain lengths). The contract or interface specification serves as the natural shared vocabulary — each agent reads it independently and generates its own internal representation.

Re-prime only after saturation. Mid-chain re-priming before the plateau (hop 3) hurts. After the plateau (hop 5+), it provides modest benefit (−0.043 CE). The threshold is empirically around hop 5 for this model and domain set. For practical architectures with 2–7 agents per level, re-priming is unnecessary — the chains never reach saturation depth.

Heterogeneous domains help. Cross-domain handoffs outperform within-domain chains. If a pipeline naturally involves different processing modes (design → implementation → testing), the domain discontinuities act as natural fences. Homogeneous pipelines (all implementation agents) accumulate more interference.

5.2 The three-level architecture

The hop scaling results validate a specific multi-agent architecture:

1. **Level 1 (Orchestrator):** Holds strategic intent. Decomposes into 2–7 components.

Operates at the architectural abstraction level. The $O(n^2)$ Graicunas limit constrains the span at this level.

2. **Level 2 (Component coordinators):** One per component. Holds the contract (shared vocabulary). Manages 2–7 competing implementations. Operates at the interface abstraction level.

3. **Level 3 (Leaf agents):** Implement. Receive fence + domain prime + task. Return pass/fail. No coordination responsibility.

The maximum chain depth in this architecture is 3 (orchestrator → component coordinator → leaf agent). The hop scaling results show that 3-hop chains degrade by ∼0.09 CE from baseline — well within the manageable range. The architecture stays shallow not because deeper chains fail but because the $O(n^2)$ coordinator limit makes wider better than deeper.

6 Updating the Throughline

Papers XXIII and XXIV update two elements of the XVI–XXII throughline.

The garbage collector hypothesis is falsified. The throughline paper (Papers XVI–XXII) proposed that the reset operates by clearing residual activation biases — garbage collection. Paper XXIII tested this directly: L2 distance to neutral does not predict coordination quality ($\rho = 0.100$, $p = 0.798$). The mechanism is mode switching, not garbage collection. The reset installs a processing mode boundary. Evidence: (1) token count negatively correlates with CE ($\rho = -0.672$), (2) reset benefit is constant (∼0.076 nats) regardless of prior context length, (3) the two stages compose independently ($\rho = 0.858$) because they target orthogonal dimensions — mode boundary and domain orientation.

Prediction 2 of the throughline is falsified. The throughline predicted that reset benefit should be proportional to prior context length ("more residual bias to clear"). Paper XXIII found the opposite: constant benefit. This is the key evidence for mode switching over garbage collection. A garbage collector should work harder with more garbage. A mode switch costs the same signal regardless.

Chain degradation does not compound. The throughline's interference interpretation predicted that coordination overhead should degrade monotonically. Paper XXIV shows it degrades logarithmically and saturates. The interference interpretation is correct about the direction but wrong about the dynamics. Interference reaches an equilibrium rather than accumulating without bound.

7 Limitations

Synthetic chains. The relay agents generate continuations, not task-directed work products. Real multi-agent chains (Pact implementations, Claude Code subagents) produce structured outputs — code, contracts, test cases. The saturation behavior may differ for structured content.

Single model. All agents in each chain are Qwen 2.5-7B. Cross-model chains (different scales, architectures) may show different degradation profiles.

Fixed intermediate length. Relay agents produce 48-token responses. Longer intermediate outputs may shift the saturation point.

Domain proxy. The four domains (medical, legal, code, science) are proxies for the processing mode diversity in real systems. Production pipelines have more nuanced mode boundaries.

8 Conclusion

Chain degradation saturates. A 10-hop sequential chain performs the same as a 5-hop chain. The context fence + domain prime protocol established in Papers XX–XXIII scales to multi-hop settings without compounding information loss.

The fence matters only at the terminal agent. Intermediate fences are overhead. Shared vocabulary substrates hurt. Cross-domain handoffs help, because domain discontinuity acts as a natural mode reset.

The binding constraint on multi-agent architecture is not information-theoretic chain degradation. It is the $O(n^2)$ coordinator complexity identified by organizational theory. Build wide and shallow — not because deep chains fail, but because the coordinator's capacity is the bottleneck, and the data says depth is free.

Data Availability

All experimental results are archived at `huggingface.co/datasets/jmcentire/paper8-data` under `paper24/`.

Series: Activation Geometry of Domain-Selective Noise Injection, Paper XXIV.