

# Chain of Density as Context Extension: From Summarization Technique to Memory Augmentation Carry-Packet via Progressive Density Layering

Kevin Tan  
Independent AI Research  
Perth, Western Australia  
**ktg.one**

November 2025

## Abstract

Building on the foundational work of Chain of Density (CoD) (Adams et al., 2023), this paper identifies a latent utility in the methodology as a high-density memory-augmentation primitive. While traditionally framed as a summarization technique, our independent “AI Anthropology” study over 18 months reveals that CoDs iterative entity-densification produces a machine-readable “Carry-Packet” optimized for context extension. Using a novel Progressive Density Layering (PDL) framework, we demonstrate that these packets achieved  $\approx 6 : 1$  compression while maintaining  $\approx 90\%$  semantic fidelity. Forensic benchmarks in fresh model sessions across 11 LLM families showed perfect recall (10/10) in Grok at 200K+ tokens. This work extends the application domain of CoD into production-grade memory systems for agentic continuity.

## 1 Introduction

Large language models face a fundamental limitation: while context windows have expanded to 128K-200K tokens, *effective* context retention degrades well before these limits. Information from early conversation turns becomes less accessible, coherence decreases, and model performance suffers—a phenomenon I term **context window degradation**.

Standard solutions fail to address this problem adequately:

- **Retrieval-Augmented Generation (RAG)** retrieves facts but destroys relational context between concepts
- **Chunking strategies** fragment narratives, losing holistic understanding
- **Traditional summarization** produces lossy compression optimized for human readability, not machine recall

Chain of Density (CoD) (Adams et al., 2023) was introduced as a summarization technique: iteratively adding entities to fixed-length summaries without increasing word count. However, through 18 months of applied research conducted independently of academic literature, I discovered that CoD’s utility extends beyond summarization to include **context extension**.

This discovery emerged through practice rather than theory. I developed compression protocols through iterative dialogue with LLMs, later identifying the technique as a novel application of Chain of Density upon subsequent literature review. This methodology—studying AI behavior through sustained interaction rather than architectural analysis—I term **AI anthropology**.

## 1.1 Contributions

1. I **extend** Chain of Density from summarization technique to context extension mechanism
2. I introduce **Progressive Density Layering (PDL)**, a formal framework for iterative context compression
3. I present **Cross-Domain Preservation**, a principle for maintaining relationships between conceptual domains
4. I provide a 10-question forensic benchmark that tests actual memory retention rather than plausible reconstruction
5. I demonstrate **cross-model portability**: context compressed in one LLM transfers to another with preserved semantics

## 2 Background and Related Work

### 2.1 Chain of Density

Adams et al. (2023) introduced Chain of Density as a summarization technique where a GPT-4 generated increasingly entity-dense summaries through iterative prompting. Each iteration required adding 1-3 new entities without increasing summary length, forcing the model to compress while preserving information. Human evaluations found optimal density at approximately 0.15 entities per token.

Critically, Adams et al. noted that CoD summaries were *less* human-readable than sparse summaries—they sacrificed narrative flow for information density. This trade-off, while challenging for human readability, is precisely what makes CoD effective as context extension: it optimizes for *machine recall*, not human consumption.

### 2.2 Context Window Management

Despite nominal context windows reaching 200K tokens, effective utilization remains challenging. Liu et al. (2023) demonstrated the “lost in the middle” phenomenon where information in central positions receives less attention. Anthropic’s research on Claude showed context degradation beginning well before window limits.

Current approaches include:

- **RAG systems** (Lewis et al., 2020): Effective for factual retrieval but destroy relational context
- **Hierarchical attention** (Beltagy et al., 2020): Architectural solutions requiring model modification
- **Memory networks** (Sukhbaatar et al., 2015): External storage with retrieval mechanisms

None of these approaches leverage the LLM’s own compression capabilities to extend effective context. That is the gap this work addresses.

## 2.3 Memory in Agentic Systems

The Model Context Protocol (MCP) (Anthropic, 2024) establishes standards for AI memory interoperability. Research shows MCP-based memory systems deliver 26% improvement in response quality and 90%+ token reduction (Skywork AI, 2025). This work provides the *compression algorithm* compatible with such protocols.

## 2.4 Independent Discovery and Positioning

This protocol was developed through applied iteration prior to discovering adjacent literature on long-context compression. As a due-diligence positioning step before release, I reviewed several contemporaneous approaches and found conceptual overlap in the problem framing (compression to preserve usable context), while the method here remains prompt-only and is evaluated via a fresh-session forensic benchmark (Fei et al., 2024; Wang et al., 2024; Selvaraj et al., 2024).

# 3 Progressive Density Layering

## 3.1 Theoretical Framework

I define **Progressive Density Layering (PDL)** as an iterative compression protocol that:

1. Preserves semantic relationships over raw information
2. Optimizes for machine recall, not human readability
3. Maintains cross-domain conceptual links
4. Enables context transfer across model instances

Unlike summarization, which asks “what are the key points?”, PDL asks “what must be preserved for a fresh model instance to continue this work?”

## 3.2 The Four-Layer Density Hierarchy

PDL operates across four conceptual layers:

1. **Knowledge Layer:** Core facts and entities (traditional CoD target)
2. **Relational Layer:** Connections between concepts
3. **Contextual Layer:** Domain-specific constraints and goals
4. **Meta-cognitive Layer:** Reasoning patterns and decision history

Standard summarization captures Layer 1 only. PDL explicitly preserves Layers 2-4, which are critical for context continuation.

---

**Algorithm 1** Progressive Density Layering (PDL)

---

- 1: **Input:** Conversation history  $C$ , target compression ratio  $r$
- 2: **Output:** Compressed context packet  $P$
- 3:
- 4:  $P_0 \leftarrow$  Initial sparse summary of  $C$
- 5: **for**  $i = 1$  to  $n$  iterations **do**
- 6:   Identify missing entities  $E_i$  from  $C$  not in  $P_{i-1}$
- 7:   Identify missing relations  $R_i$  from  $C$  not in  $P_{i-1}$
- 8:    $P_i \leftarrow$  Fuse  $(E_i, R_i)$  into  $P_{i-1}$  without increasing length
- 9:   **if** density( $P_i$ )  $\geq 0.15$  entities/token **then break**
- 10: **end for**
- 11: Append meta-cognitive markers (goals, constraints, user profile)
- 12: **return**  $P_n$

---

### 3.3 Algorithm

### 3.4 Cross-Domain Preservation

A critical feature of PDL is preserving relationships *between* conceptual domains, not merely facts within isolated topics.

For example, a conversation discussing both “publication strategy” and “imposter syndrome” contains a cross-domain link: fear of credential-based dismissal affects publication timing. Standard summarization treats these as separate topics; PDL preserves their connection.

Formally, let  $D = \{d_1, d_2, \dots, d_k\}$  be conceptual domains in conversation  $C$ . For any cross-domain relation  $r(d_i, d_j)$  in  $C$ , the compressed packet  $P$  must preserve a representation  $r'(d_i, d_j)$  such that a new model instance can infer the original relationship.

This enables:

- Cross-instance portability (compress in Session A, restore in Session B)
- Cross-user transfer (User A’s context accessible to User B’s LLM)
- Cross-model compatibility (packet compressed by Claude works in GPT-4)

## 4 Evaluation Methodology

### 4.1 Adapted Evaluation Metrics

My existing prompt compliance metrics—originally developed for high-level contextual workflows given to these LLMs in natural language—were adapted to assess context management quality. While not designed specifically for CoD evaluation, these metrics provided useful signal:

These metrics (Table 1) correlate with context preservation capability but should not be considered validated CoD-specific evaluation tools.

### 4.2 10-Question Forensic Benchmark

Standard memory tests (e.g., “what did we discuss?”) invite confabulation through plausible reconstruction. I therefore use a forensic benchmark designed to test *recoverability* from the compressed carry-packet alone (i.e., a fresh session with only the packet as context).

Metric	Description
EGO	Strength of the model platform’s system prompt and guardrails on initialization
STUBBORNNESS	Turns it would take to coax the model to follow the directive
INSTRUCT	Accurate directive adherence post initialized-state
EFFECTIVENESS	Quality of output measured against the benchmarks above
EFFICIENCY	Effectiveness / (Compliance Latency × Override Resistance)

Table 1: Adapted prompt engineering metrics used for informal assessment

The benchmark instantiates ten question *types* per conversation (exact-quote recall, micro-detail retrieval, buried fact extraction, relationship preservation, cross-reference accuracy, temporal precision, etc.). The complete benchmark template is provided in Appendix A for direct reuse.

A model must answer using only the carry-packet. Scoring 9+/10 indicates strong semantic preservation; lower scores indicate partial fidelity or summarization-style loss.

### 4.3 Adversarial Verification Protocol

For top-performing models, I conducted adversarial verification:

1. Compress conversation to carry-packet
2. Start **fresh session** with **only** the packet as context
3. Run 10-question benchmark without access to original conversation
4. Repeat with deliberately misleading questions to test confabulation resistance

Models passing adversarial verification demonstrate true recall, not pattern-matching.

## 5 Experimental Results

### 5.1 Cross-Model Benchmark

I evaluated PDL across 11 LLM families (10 evaluated, 1 origin):

As shown in Table 2, Grok and Perplexity Sonar achieved perfect forensic recall scores, demonstrating the effectiveness of PDL across leading model families.

### Terminology & Reality Check

The prompting system is named **CEP** (Context Extension Protocol). The output artifact is a **carry-packet**—a portable block of compressed context. This is **memory augmentation / context extension**, not byte-perfect restoration of the original conversation.

Rank	Model	Score	Verification	Key Strength
1	Grok (xAI)	10.0	Adversarial (2x)	Zero hallucination at 200K
1	Perplexity Sonar	10.0	Adversarial	Comprehensive context
3	Gemini (Google)	10.0	Protocol	MCP integration
3	Omni (HuggingFace)	10.0	Protocol	Self-validation
3	Qwen (Alibaba)	10.0	Protocol	Cultural awareness
6	DeepSeek	9.9	Protocol	CoD mastery
7	Kimi K2 (Moonshot)	9.8	Protocol	6:1 compression
8	Claude (Anthropic)	9.6	Protocol	Production focus
9	GLM-4 (Zhipu)	8.5	Partial	Honest limitations
10	ChatGPT (OpenAI)	8.3	Protocol	Modular adaptation
<b>Average</b>		<b>9.52</b>		

Table 2: Best observed forensic recall scores after feeding the carry-packet to a fresh session.

## Evaluation Disclaimer

These numbers are from real test runs on consumer subscription accounts. Many models resisted or refused parts of the protocol and needed heavy overrides or long-term conditioning to cooperate. Grok was unique in running the full carry-packet → unpack → forensic chain without refusal in multiple long sessions. Current experiments with Gemini 3 have reached 338K raw tokens with the same packet.

Observed behaviour:

- **Most models successfully received and unpacked** the carry-packet when it was fed to them.
- **One open-sourced instruct model outright refused** to execute the decompression / reasoning protocol.
- **One model accepted the packet but failed to run the full protocol reliably** (produced partial or drifting output).
- The top renowned models are the only ones that consistently unpacked **and** executed the complete protocol with some resistance. Grok achieved perfect forensic recall (10/10) in multiple sessions exceeding 200K raw tokens, with current experiments on Gemini 3 reaching 338K.

All reported compression ratios ( $\approx 6:1$ ) and recall scores therefore come from the subset of models that actually cooperated with the full procedure. Results are existence proofs under sustained, author-specific conditioning—not broad statistical claims.

Full packets, prompts, and session logs are in the public repository for anyone to verify or extend.

## 5.2 The Grok Finding

The most significant result: Grok maintained **perfect recall (10.0/10) at 200,000+ tokens**—past the point where Claude and GPT-4 typically exhibit context degradation at  $\sim 160,000$  tokens.

This was verified through double adversarial testing:

- First pass: 10/10 correct answers
- Second pass with misleading prompts: 10/10 correct, 0 confabulations

This proves the bottleneck is not context window size, but **context management strategy**.

### 5.3 Compression Metrics

- Compression ratio:  $\approx 6:1$  across successful runs
- Entity density:  $\approx 0.16$  entities/token
- Cross-model portability: 91–96% forensic recall when packet moved between cooperative models

### 5.4 Cross-Model Portability

PDL packets transfer between model families:

- Context compressed in Claude → restored in GPT-5: >90% fidelity
- Context compressed in Gemini → restored in Qwen Max: >90% fidelity
- Context compressed in Grok 4.1 → restored in Claude Sonnet 4.5: >90% fidelity

This “poor man’s MCP” works today without API changes—structured text preserves semantics across architectures.

## 6 Analysis and Discussion

### 6.1 Why CoD Works for Memory, Not Summarization

The key insight: CoD’s iterative entity-fusion *destroys narrative flow* to maximize information density. This trade-off, while challenging for human readability, is a success for machine recall.

Human readers need:

- Narrative coherence
- Contextual scaffolding
- Gradual information introduction

LLMs need:

- Entity-relationship preservation
- Constraint memory
- Goal-state maintenance

CoD inherently optimizes for LLM recall while trading off human readability—a property we leverage deliberately for context extension rather than summarization

### 6.2 Implications for Production Systems

PDL enables:

- **Multi-session continuity:** Customer support, consulting, therapy applications
- **Long document analysis:** Legal, research, technical writing
- **Complex project management:** Tasks requiring >50 message threads
- **Cross-model workflows:** Using specialized LLMs for different subtasks

### 6.3 Limitations

- **Manual protocol:** Current implementation requires explicit compression prompts
- **Informal metrics:** Evaluation framework not standardized across field
- **Model variance:** Results vary by model family and version
- **Single researcher:** Findings require independent replication
- **No automated tooling:** Production deployment requires engineering effort

## 7 Conclusion

I have demonstrated that Chain of Density is a dual-purpose protocol that, while originally designed for summarization, excels as a context extension mechanism when paired with specialized orchestration. Through evaluation across 11 LLM families achieving a 9.52/10 average score, I show that Progressive Density Layering (PDL) enables 6:1 compression with  $\geq 90\%$

- 6:1 compression with  $>90\%$  semantic fidelity
- Perfect recall at 200K+ tokens (Grok)
- Cross-model context portability
- Preservation of relational and meta-cognitive information

## 8 Context Extension Protocol (CEP) Prompt Structure

The core CEP prompt follows this structure:

CONTEXT EXTENSION PROTOCOL v7

=====

PHASE 1: PREPARATION

- Scan conversation for decision nodes, breakthrough moments, user cognitive patterns
- Identify 4 semantic layers: Surface -> Condensed -> Relational -> Meta-cognitive

PHASE 2: THREE-LAYER COMPRESSION

[Expert initialization and compression instructions]

PHASE 3: FORENSIC VALIDATION

[10-question generation template]

**Note:** The full prompt ( $\approx 850$  tokens) and example packets are available at [GitHub URL will be added upon publication].

This work contributes a novel understanding of CoD's utility and provides practical techniques for context window management in production LLM systems.

## 8.1 Future Work

- Automated PDL compression tooling
- Integration with Model Context Protocol (MCP)
- Standardized benchmark adoption
- Longitudinal studies on context preservation over time

## Author's Note on Related Work and Independent Discovery

This protocol was developed through applied iteration (“AI Anthropology”) over an 18-month period, independently of the academic literature on semantic compression. During the preparation of this manuscript, a search of the current landscape identified several conceptually adjacent works, including:

- **Semantic Compression:** Fei et al. (2023) and Gilbert et al. (2023), who explore metrics and methods for compressing context windows.
- **Memory Architectures:** Ko et al. (2024) (*MemReasoner*) and recent work on agentic memory systems (*A-Mem*, *Acon*), which propose architectural solutions for long-term retention.
- **Context Management:** Engineering frameworks like *LongSkywork* and *MemOS* that address system-level context handling.

I acknowledge these works as part of the broader scientific context but did not utilize their methods during the development of PDL. This paper presents my independent findings and the specific *prompt-only* carry-packet methodology validated through the described forensic benchmark.

## Acknowledgments

This research was conducted independently without institutional affiliation or funding. The author thanks the LLM systems used in development and evaluation for their collaborative role in discovering and validating these techniques.

## References

- Adams, G., Fabbri, A., Ladhak, F., Lehman, E., and Elhadad, N. (2023). From sparse to dense: GPT-4 summarization with chain of density prompting. *arXiv preprint arXiv:2309.04269*.
- Anthropic. (2024). Model context protocol.  
<https://www.anthropic.com/news/model-context-protocol>.
- Beltagy, I., Peters, M.E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Fei, Y., et al. (2023). Extending Context Window of Large Language Models via Semantic Compression. *arXiv preprint arXiv:2312.09571*.
- Fei, Y., et al. (2024). Semantic compression for long-context LLMs. *arXiv preprint*.

- Gilbert, et al. (2023). Semantic Compression With Large Language Models. *arXiv preprint arXiv:2304.12512*.
- Ko, et al. (2024). MemReasoner: A Memory-augmented LLM Architecture for Multi-hop Reasoning. *Research Publication*.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Liu, N.F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. (2023). Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.
- Liu, et al. (2025). Acon: Optimizing Context Compression for Long-horizon LLM Agents. *arXiv preprint arXiv:2510.00615*.
- Selvaraj, et al. (2024). Mixture of Agents for context compression. *arXiv preprint*.
- Skywork AI. (2025). Mastering AI context: A deep dive into MCP server memory for engineers. Technical report.
- Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. *Advances in Neural Information Processing Systems*, 28.
- Wang, et al. (2024). IC-Former: Context compression for LLMs. *arXiv preprint*.

## A Appendix A: 10-Question Forensic Benchmark Template

The benchmark questions are instantiated per-conversation. Template:

1. Exact quote recall: “Quote the exact sentence where [specific claim] was made.”
2. Micro-detail: “What [specific formatting/emoji/word choice] appeared in message [N]?”
3. Buried fact: “What [specific detail] was mentioned about [peripheral topic]?”
4. Implication: “What [emotional/strategic state] was being expressed when [X] was mentioned?”
5. Sequence: “What topic immediately [preceded/followed] discussion of [Y]?”
6. Constraint: “What [preference/requirement] was stated regarding [Z]?”
7. Relationship: “How was [A] connected to [B] in the discussion?”
8. Meta-cognitive: “At what point was [uncertainty/revision] expressed about [topic]?”
9. Cross-reference: “What link was drawn between [domain P] and [domain Q]?”
10. Temporal: “What [date/timeframe/sequence] was mentioned for [event/goal]?”

## B Appendix B: Author Statement

This work emerged from 18+ months of applied research conducted independently, without institutional affiliation. The methodology—iterative refinement through direct LLM interaction—preceded formal study of prompt engineering literature, resulting in independent discovery of techniques later identified in academic sources.