Scaling Episodic Compression: A Two-Layer Architecture for Continuous Learning

Part 2 of Learning Through Narratives: Episodic Compression to Latent Variables for Sample-Efficient Intelligence

Terminology Note: As in Part 1, we use terms like "learning," "understanding," and "reasoning" as behavioral shorthand without making claims about machine consciousness or internal mental states. When we say an AI system "learns" from narratives, we mean it produces behavioral responses consistent with narrative patterns. This usage aligns with our focus on behavioral competence rather than unverifiable internal states. Similarly, we use "latent variables" in the standard ML sense of learned hidden representations that capture underlying structure, distinct from LeCun's recent emphasis on continuous latent variables for handling aleatory uncertainty in predictions.

Abstract

Part 1 demonstrated that episodic compression enables sample-efficient learning through experimental validation in an artificial world. A system with 9 generalizations and 4 episodes successfully predicted outcomes in novel scenarios. However, this validation occurred at what we term "toy scale" – where all knowledge fits comfortably within a single language model's context window. At this scale, the proposed episodic architecture proved unnecessary; a single model handled pattern-matching, reasoning, and consolidation without specialized support.

This observation suggests the architecture's value emerges specifically at scale, when accumulated world models exceed single-context capacity. We propose a two-layer design addressing this scaling challenge through functional separation. The Intuition layer maintains generalization storage and performs rapid pattern-matching using small models with large contexts. The Executive layer conducts deliberate reasoning and consolidation through frequent "micro-sleep" cycles using large capable models with standard contexts.

This separation appears analogous to distinctions in cognitive neuroscience between automatic pattern recognition and controlled reasoning processes (Schneider & Shiffrin, 1977; Kahneman, 2011), though our computational implementations differ from biological substrates. The architecture enables world models to scale beyond single-context limitations while maintaining learning efficiency through existing pre-trained language models.

We provide implementation guidance and architectural specifications while acknowledging this work remains conceptual – comprehensive empirical validation at scale awaits future research.

1. Introduction

1.1 From Validation to Scaling Challenges

Part 1 established that curiosity-driven episodic compression may achieve sample-efficient learning without embodiment. The Shimmer Valleys experiment validated core mechanisms: salient experiences compress into episodes, episodes generate generalizations through pattern detection, and compact generalization sets enable prediction in novel scenarios. A system with 9 generalizations and 4 episodes successfully reasoned about five complex scenarios, demonstrating appropriate confidence calibration and multi-step causal inference.

This validation proved the principle could work. It also revealed something unexpected: at this scale – 9 generalizations, 4 episodes, approximately 500 words total – specialized episodic machinery provided no measurable advantage. A single language model simultaneously held all generalizations, matched patterns against current experiences, and performed deliberate reasoning without architectural support. The episodic compression framework provided conceptual organization, but the implementation required no separation between storage, retrieval, and reasoning.

Real-world continuous learning appears to operate at different scale. A therapist's accumulated knowledge might span thousands of case patterns, theoretical frameworks, and interaction dynamics. A domain expert maintains vast networks of causal relationships, experimental results, and theoretical principles. These world models would substantially exceed current context window capacities, even with extended contexts of 100K-200K tokens.

The question becomes: at what point does specialized architecture become necessary, and what form should it take?

1.2 The Scaling Transition Point

Consider how knowledge accumulation scales. A well-formed generalization requires approximately 50-100 tokens to encode its causal structure, scope, confidence level, and supporting evidence. A retained high-salience episode requires approximately 100-150 tokens for situation description, unexpected outcomes, and salience markers.

For an 8K token context window with 2K tokens reserved for current experience and reasoning, available storage supports approximately 60-120 generalizations or a mixed set of roughly 50 generalizations plus 30 episodes. A 32K context window with similar reservations might accommodate 200-300 generalizations. Extended contexts of 200K tokens could theoretically hold 1,500+ generalizations.

These calculations suggest several transition points where single-model approaches may become insufficient:

Toy scale (< 50 generalizations): Single model with standard context appears sufficient. No architectural support needed beyond prompt engineering.

Moderate scale (50-200 generalizations): Single model with extended context may manage, though retrieval efficiency and consolidation complexity increase.

Real-world scale (200-1,000+ generalizations): Even extended contexts face challenges beyond mere token capacity. Attention mechanisms distribute across all tokens in context – with 1,000+ generalizations present, attention weights spread thinly and pattern-matching precision may degrade. Consolidation processes that extract patterns from hundreds of accumulated episodes grow superlinearly complex.

Enterprise scale (1,000+ generalizations): Systems operating continuously for months or years would accumulate knowledge exceeding any single context. Functional separation appears necessary.

However, these thresholds remain speculative. We lack empirical evidence for where transitions occur or whether they occur sharply or gradually. The analysis provides estimates pending experimental validation.

1.3 Parallels with Cognitive Architecture

Research in cognitive neuroscience has identified distinctions between automatic and controlled processing systems (Schneider & Shiffrin, 1977; Shiffrin & Schneider, 1977). Automatic processing operates rapidly on familiar patterns with minimal attentional resources. Controlled processing handles novel situations through deliberate reasoning at greater metabolic cost. This functional separation appears across multiple brain systems – motor control (cerebellum versus motor cortex), memory (procedural versus declarative systems), and perception (bottom-up versus top-down attention).

Kahneman (2011) popularized these distinctions as System 1 (fast, automatic, parallel) versus System 2 (slow, deliberate, serial). While his framework addresses reasoning and decision-making rather than memory systems specifically, the computational properties align: automatic processes handle pattern recognition efficiently across large knowledge bases, while controlled processes engage for complex reasoning within limited working memory.

We observe that episodic compression at scale might benefit from analogous functional separation, though our implementations use language model substrates rather than biological neural networks. Fast pattern-matching across large generalization stores appears computationally distinct from deliberate reasoning about novel situations. Separating these functions may address the scaling challenges identified above.

We make no claims that this architecture mirrors how biological cognition works, only that similar functional separations may prove useful for different computational reasons.

1.4 The Two-Layer Proposal

We propose addressing scaling through functional separation into two components:

Intuition maintains consolidated knowledge and performs pattern-matching. It stores all generalizations and select high-salience episodes, operates with large context windows (100K-200K tokens) to enable comprehensive pattern-matching, uses small efficient models (1-3B parameters) for rapid inference, and runs continuously in parallel with Executive processing.

Executive handles reasoning and consolidation. It performs deliberate analysis of current situations, works with standard context windows (8K-32K tokens) processing only relevant subsets retrieved from Intuition, conducts periodic consolidation during "micro-sleep" cycles, uses larger capable models for complex reasoning, and engages selectively when Intuition flags novelty or users initiate complex queries.

This separation exploits different computational properties: Intuition leverages transformer attention mechanisms for parallel pattern-matching across extensive context. Executive performs serial depth-first reasoning within limited context. Both components use language models, but optimized for different computational demands.

The architecture resembles retrieval-augmented generation (RAG) superficially but differs fundamentally in mechanism, as we discuss in Section 5. RAG uses embedding-based semantic similarity to retrieve text chunks. Intuition uses full language model attention to pattern-match causal structure. RAG retrieves from static knowledge bases. This architecture learns through consolidation cycles.

1.5 Key Claims and Non-Claims

Our thesis comprises several claims:

1. Episodic compression architecture becomes necessary at scale when world models exceed single-context capacity
2. Functional separation into pattern-matching (Intuition) and reasoning (Executive) may address scaling challenges
3. The separation appears analogous to cognitive architecture distinctions, though implementations differ
4. Small models with large contexts can perform effective pattern-matching across extensive knowledge bases
5. Frequent brief consolidation cycles may suffice for continuous learning
6. The approach requires no new neural network training, only architectural design

We explicitly do not claim:

- That this is how biological cognition works (only that functional separations may be useful)
- That we have empirically validated this architecture at scale (comprehensive testing awaits)
- That thresholds and parameters are optimal (they require empirical tuning)
- That this approach is universally superior to alternatives (different use cases may favor different solutions)

- That the implementation details we provide are the only or best way to realize these principles

1.6 Paper Organization

Section 2 analyzes scaling behavior and identifies where specialized architecture becomes valuable. Section 3 presents the two-layer design including information flow and consolidation mechanisms. Section 4 provides implementation specifications. Section 5 compares this approach with alternatives including RAG and LeCun's JEPA framework. Section 6 addresses limitations and future work. Section 7 concludes.

2. When Does Architecture Matter?

2.1 Empirical Observations from Part 1

The Shimmer Valleys experiment operated at toy scale: 9 generalizations, 4 episodes, approximately 500 words total. A single language model with standard 8K context window could simultaneously:

- Hold all generalizations in working context
- Hold all retained episodes
- Process incoming test scenarios
- Match patterns against stored knowledge
- Perform multi-step causal reasoning
- Generate predictions with confidence calibration

No specialized architecture was required or would have provided measurable benefit. The model's native attention mechanisms handled pattern-matching. Its reasoning capabilities applied directly to prediction tasks. The episodic compression framework provided useful conceptual organization, but implementation needed no separation between storage, retrieval, and reasoning.

This reveals an important insight: the architecture's value emerges specifically when world models exceed what single contexts can handle effectively.

2.2 Token Capacity Analysis

We can estimate transition points through context window capacity analysis, though we acknowledge these remain speculative pending empirical validation.

A well-formed generalization encoding causal structure, scope, confidence, and supporting evidence requires approximately 50-100 tokens. For example:

"When whisps form choruses, they transform surfaces beneath them. Over living glass → turns purple and solid. Over shade pools → pool overflows creating mirror surfaces. Confidence:

HIGH (5 supporting episodes: E2, E7, E12, E18, E23). Scope: Applies to all whisp choruses; no known exceptions."

This format captures the essential pattern in approximately 75 tokens.

A retained high-salience episode with situation description, unexpected outcomes, and salience markers requires approximately 100-150 tokens:

"Resonator struck → rang out → whisps converged forming chorus above → living glass beneath turned purple and solid → glob rolled onto purple surface and became permanently purple. Salience: HIGH (unexpected permanent state change). Context: Equipment interaction trial, Day 3."

For different context window sizes with tokens reserved for current experience and reasoning:

8K context (2K reserved):

- Available storage: 6K tokens
- Capacity: ~60-120 generalizations OR ~40-60 episodes OR ~50 generalizations + 30 episodes

32K context (8K reserved):

- Available storage: 24K tokens
- Capacity: ~240-480 generalizations OR ~160-240 episodes OR ~200 generalizations + 100 episodes

200K context (20K reserved):

- Available storage: 180K tokens
- Capacity: ~1,800-3,600 generalizations OR ~1,200-1,800 episodes OR ~1,500 generalizations + 800 episodes

These calculations suggest potential transition points, though actual thresholds may differ based on factors beyond token counts.

2.3 Factors Beyond Token Capacity

Context window capacity alone does not determine when specialized architecture becomes valuable. Other factors emerge at scale:

Attention dilution: Transformer attention mechanisms distribute across all tokens in context. With 1,000+ generalizations present, attention weights spread thinly. Pattern-matching may become less precise as relevant signals dilute across vast stored knowledge. However, we lack empirical evidence for whether or when this becomes problematic.

Consolidation complexity: Extracting patterns from 100+ accumulated episodes requires comparing across many possible combinations, identifying commonalities, checking consistency. Processing overhead appears to grow superlinearly with episode count, though actual scaling behavior requires measurement.

Real-time constraints: Users expect responsive interaction. If pattern-matching and consolidation introduce multi-second latencies, systems become impractical. Separation might allow background consolidation without blocking interaction, though buffering and async processing could address this without architectural separation.

Cost considerations: Running large capable models continuously for both pattern-matching and reasoning incurs substantial costs. Using small efficient models for pattern-matching with selective engagement of large models for reasoning might reduce operational costs significantly, though actual economics depend on usage patterns.

These factors suggest architecture becomes valuable before absolute token limits are reached, but the specific transition points remain uncertain.

2.4 Tentative Threshold Estimates

Based on token capacity analysis and qualitative factors, we estimate (with low confidence) that specialized architecture may become beneficial:

With standard contexts (8K-32K): Around 50-100 generalizations, where token capacity becomes constraining and consolidation complexity increases.

With extended contexts (100K-200K): Around 200-400 generalizations, where attention dilution and consolidation overhead may emerge as bottlenecks even though token capacity remains available.

For any context size: When continuous operation over months or years would accumulate knowledge exceeding practical context limits.

These estimates provide starting points for experimentation. Actual thresholds likely vary by domain complexity, consolidation frequency, model capabilities, and other factors requiring empirical investigation.

The critical insight remains: episodic compression architecture provides value specifically when scaling beyond single-context capacity, not at toy scales where simpler approaches suffice.

3. The Two-Layer Architecture

3.1 Design Principles

The architecture separates two computationally distinct functions that become valuable at scale:

Fast pattern-matching across extensive knowledge stores, identifying which stored generalizations relate to current experiences. This leverages transformer attention mechanisms for parallel processing across large contexts.

Deliberate reasoning about current situations, combining retrieved patterns with flexible thinking to handle novel scenarios. This requires depth and capability within limited working context.

Attempting both simultaneously in a single model at scale faces challenges: either working context becomes overcrowded with stored knowledge reducing reasoning capacity, or knowledge bases must be truncated losing comprehensiveness. Separation allows each function to use computational resources optimally.

We emphasize this represents one possible approach among alternatives, not necessarily the optimal solution for all use cases.

3.2 Component Specifications

Intuition Component:

Purpose: Maintains world model and performs pattern-matching retrieval

Implementation:

- Small language model (1-3B parameters; examples: Phi-1.5, Phi-2, Gemma-2B)
- Large context window (100K-200K tokens)
- Continuous operation in parallel with Executive
- Low computational cost per inference

Storage:

- All consolidated generalizations (abstract causal patterns)
- Highly salient episodes (exceptional specific cases, ~10% of created episodes)

Primary functions:

- Store accumulated knowledge (generalizations and exceptional episodes)
- Pattern-match current experiences against stored knowledge via attention mechanisms
- Detect novelty when no patterns match adequately
- Surface relevant patterns (generalizations and episodes) to Executive

Selection rationale: Small models provide faster inference and lower costs while large contexts enable comprehensive pattern-matching across extensive knowledge bases. The dual storage of generalizations and episodes mirrors biological distinctions between semantic memory (general knowledge) and episodic memory (specific events), both contributing to pattern recognition.

Executive Component:

Purpose: Performs reasoning and consolidates learning

Implementation:

- Large capable language model (20B-100B+ parameters; examples: Claude Opus, GPT-4)
- Standard context window (8K-32K tokens)
- Selective engagement triggered by novelty or user queries
- Higher computational cost, used judiciously

Primary functions:

- Deliberate reasoning about current situations
- Generate predictions combining patterns with flexible thinking
- Multi-step causal inference and explanation
- Periodic consolidation extracting patterns from episodes
- Consistency checking across world model

Selection rationale: Large capable models provide reasoning depth and flexibility while standard contexts suffice since Executive works with retrieved relevant subsets rather than full knowledge bases.

3.3 Information Flow During Interaction

Real-time interaction involves parallel processing. When a user query or new experience arrives:

```
User Experience
  ↓
  ├──→ Intuition: Pattern-match against stored knowledge
  │       - Scan generalizations for structural matches
  │       - Surface relevant patterns or report "no match"
  │       - Flag novelty if patterns inadequate
  │
  └──→ Executive: Receive experience + pattern matches
          - Access raw experience details
          - Access relevant generalizations from Intuition
          - Perform deliberate reasoning combining both
          - Generate response with predictions if needed
          - Detect prediction mismatches (salience marker)
          - Flag high-salience episodes for consolidation
          ↓
          Return to user
```

Key aspects of this flow:

Parallel processing: Intuition and Executive both receive experiences simultaneously. Intuition performs rapid pattern-matching while Executive prepares for reasoning. Results combine before response generation.

Pattern-matching precedes reasoning: Intuition identifies relevant generalizations before Executive engages, narrowing focus to applicable patterns rather than requiring search through thousands of stored generalizations.

Salience detection distributed: Intuition detects novelty when no patterns match. Executive detects prediction failures when outcomes violate expectations. Both can flag experiences for consolidation.

Response generation in Executive: Final responses come from Executive, which accesses both raw experience and Intuition's pattern matches. This enables coherent reasoning that can reference specific generalizations when explaining predictions.

We note this design represents one possible approach – alternatives might use different information flows or triggering conditions.

3.3.1 Temporal Trace Management

Every episode and generalization transmitted between the Intuition and Executive layers includes an explicit timestamp field. This serves several purposes across the two-layer architecture:

Recency weighting. Intuition can favor more recent experiences when matching patterns, allowing adaptive attention to current conditions without discarding older knowledge.

Temporal consistency during consolidation. The Executive can evaluate how long it has been since a generalization was last updated, triggering review or refinement based on elapsed time rather than fixed batch size.

Order-aware reasoning. Because timestamps travel with retrieved artifacts, the Executive can determine whether an observed relation is merely coincident or genuinely sequential, grounding causal reasoning in temporal evidence.

For episodes, the timestamp records when the system experienced the event. For generalizations, the timestamp records when the pattern was formed or last updated during consolidation.

This approach achieves temporal grounding with minimal complexity. The architecture never constructs an explicit event graph; it simply relies on consistent timestamps to preserve the ordering and spacing of experiences, ensuring that time itself becomes an accessible dimension of reasoning.

3.4 Consolidation Through "Micro-Sleep" Cycles

Rather than mimicking human 8-hour sleep cycles (which may reflect circadian rhythms and metabolic constraints beyond pure computational requirements), the architecture proposes frequent brief consolidation periods. This remains speculative – optimal consolidation timing requires empirical investigation.

Trigger conditions:

Primary: Episode accumulation (suggested 5-10 high-salience episodes) Secondary: Time-based fallback (suggested 2-6 hours maximum)
Tertiary: Opportunistic during idle periods

Consolidation process:

Phase 1: Executive loads flagged high-salience episodes (typical batch: 5-15 episodes)

Phase 2: Executive queries Intuition for related historical episodes, receiving pattern-matched context from archives

Phase 3: Executive analyzes combined episode set identifying commonalities and causal structure

Phase 4: For each identified pattern:

- Check if similar generalization exists
- If yes: update confidence, refine scope, or split into conditional variants
- If no and supporting episodes ≥ threshold (suggested 3-5): form new generalization

Phase 5: Consistency checking across new and existing generalizations, identifying contradictions requiring resolution

Phase 6: Write new/updated generalizations back to Intuition's store, mark episodes as consolidated

Duration and frequency:

Typical duration: 10-60 seconds depending on batch size Typical frequency: 3-8 times per day during active learning periods User experience: Brief unavailability with system message

These parameters remain tentative pending empirical validation across different domains and usage patterns.

3.5 Why This Separation May Help at Scale

The two-layer architecture potentially addresses scaling challenges identified in Section 2:

Context limitations: Intuition holds full generalization store in large context while Executive works with retrieved subsets, enabling world models beyond single-context capacity.

Retrieval efficiency: Small model pattern-matching provides sub-second latency while parallel processing eliminates user-visible delays.

Real-time constraints: Pattern-matching happens in parallel without blocking. Consolidation runs in background during micro-sleeps without disrupting interaction.

Coherence maintenance: Consolidation includes explicit consistency checking, potentially maintaining world model coherence as knowledge accumulates.

Computational cost: Small model runs continuously at low cost while large model runs selectively, potentially providing favorable cost scaling.

However, these benefits remain theoretical pending empirical validation. Alternative approaches might address the same challenges differently.

4. Implementation Considerations

4.1 Model Selection Criteria

For Intuition Component:

Context window size appears most critical – must support 100K+ tokens minimum for substantial knowledge bases. Current options include Claude Haiku (200K), Gemini Pro (1M), GPT-4-Turbo (128K).

Attention quality matters for pattern-matching precision. Models should reliably identify structural similarities in causal patterns. This requires testing with domain-specific matching tasks.

Inference speed affects user experience since Intuition runs on every interaction. Sub-second response time appears preferable.

Per-token cost accumulates since Intuition runs continuously. Smaller models with large contexts (e.g., Phi-1.5 extended, if available) may provide optimal economics.

For Executive Component:

Reasoning capability appears most critical – must handle novel situations requiring flexible thinking, perform multi-step causal inference, and extract patterns from episode sets. Current options include Claude Opus, GPT-4, Gemini Ultra.

Consistency checking matters for consolidation quality. Models should reliably detect contradictions and reason about conditional variants.

Instruction following affects consolidation reliability. Models should adhere to protocols and produce structured outputs.

Per-invocation cost accumulates since consolidation runs periodically. More capable models may justify costs through better consolidation quality, though this requires validation.

4.2 Storage Format Considerations

Generalizations might encode as structured text preserving human readability:

```
<generalization id="G1" confidence="high" sources="E1,E4,E7,E12">
Pattern: Contact with colored globs transfers color through chains
with decreasing intensity
Scope: Applies to normal globs; exceptions include phase globs, glowing globs
Evidence: 7 supporting episodes showing consistent behavior
Last Updated: 2025-03-15
</generalization>
```

This format encodes causal patterns, scope limitations, confidence levels, and evidence trails while remaining interpretable.

Episodes similarly might use structured natural language:

```
<episode id="E1" salience="8" timestamp="2025-03-10" status="consolidated">
Situation: Player touched blue glob
Outcome: Player's hand turned blue; nearby gray glob touched hand and turned
blue
Surprise: Unexpected color transfer through chain contact
Causal Structure: Entity(property:blue) + Contact → Transfer(property:blue) +
Chain
Supporting: G1
</episode>
```

Alternative formats (JSON, structured databases) might provide computational benefits while sacrificing readability. Format selection likely depends on specific use case priorities.

4.3 Pattern-Matching Implementation

Intuition's pattern-matching may leverage language models' native attention mechanisms rather than requiring specialized retrieval algorithms. When Intuition receives current experience with full generalization store in context, attention mechanisms naturally compute relevance across stored patterns.

For example, given experience "chrome flutter seed touches resonator" with generalizations about contact, property transfer, and sound emission in context, multi-head attention might match on entity types, properties, actions, and causal structures simultaneously. The model generates output identifying relevant generalizations with relevance scores.

This approach requires no external retrieval system – pattern-matching emerges from standard language model operation. Small model size keeps this fast while large context enables comprehensive matching.

However, this mechanism requires empirical validation. Attention-based pattern-matching may degrade at very large scales or models may require specific prompting strategies to perform structural rather than semantic matching.

4.4 Consolidation Implementation

Executive consolidation may operate through structured prompting rather than specialized learning algorithms:

```
You are conducting consolidation to extract patterns from recent experiences.

HIGH-SALIENCE EPISODES (new):
[Batch of 5-10 flagged episodes]

RELATED HISTORICAL EPISODES (from Intuition):
[3-10 similar episodes retrieved by pattern-matching]

EXISTING GENERALIZATIONS:
[Current generalization store]

TASK:
1. Identify common causal patterns across episodes
2. Determine if existing generalizations apply
3. Form new generalizations if ≥3 episodes share consistent pattern
4. Update confidence levels based on evidence
5. Check for contradictions requiring resolution

OUTPUT: [Structured generalization updates]
```

Executive processes this using native reasoning capabilities – no gradient descent or weight updates occur. Consolidation happens through inference over structured prompts.

This approach leverages pre-trained capabilities (causal reasoning, consistency checking, conditional reasoning, confidence calibration) without requiring training. However, prompt engineering quality likely affects consolidation effectiveness significantly.

4.5 Deployment Considerations

Consolidation scheduling: Monitor salience accumulation and trigger at thresholds (suggested 5-10 episodes). Use time-based fallback (suggested 2-6 hours maximum). Consider opportunistic consolidation during idle periods.

Context management: Periodically refresh Intuition's context to prevent drift. Compose Executive context from current experience, pattern matches, and retrieved relevant generalizations within window limits.

Cost optimization: Use most cost-efficient small model for Intuition's continuous operation. Reserve large capable model for Executive's selective engagement. Batch consolidation to amortize costs.

Monitoring: Track episodes per day, generalizations formed, consolidation duration, and retrieval accuracy. Alert on contradictions, consolidation failures, context approaching limits, or quality degradation.

These represent starting points requiring adjustment based on actual deployment experience and usage patterns.

5. Relationship to Other Approaches

5.1 Comparison with LeCun's JEPA Framework

Part 1 discussed conceptual differences between episodic compression and Joint Embedding Predictive Architectures (LeCun, 2022). Here we focus on practical implementation differences:

Learning source: JEPA trains on embodied interaction or video sequences. Episodic compression leverages narrative grounding from pre-training.

Training requirements: JEPA trains hierarchical networks from scratch requiring weeks on specialized hardware. Episodic compression uses pre-trained models through architectural design enabling immediate deployment.

Latent variables: JEPA optimizes representations through gradient descent minimizing prediction error. Episodic compression generates representations through salience-based compression without parameter updates.

Domain adaptation: JEPA requires retraining or fine-tuning for new domains. Episodic compression requires new narrative curation or experiences without retraining.

These approaches appear to address similar problems (sample-efficient learning, world model construction) through different mechanisms. For embodied robots manipulating physical objects, JEPA's sensorimotor grounding may prove more suitable. For language-based agents reasoning about abstract domains, episodic compression's narrative grounding may prove more practical.

We view these as complementary contributions rather than competing alternatives.

5.2 Distinction from Retrieval-Augmented Generation

Intuition superficially resembles RAG but differs fundamentally in mechanism:

Standard RAG: Stores text chunks, retrieves via embedding similarity, maintains static knowledge bases, lacks novelty detection, performs retrieval only without learning.

Intuition: Stores structured generalizations encoding causal patterns, retrieves via attention-based pattern-matching on structure, updates dynamically through consolidation, explicitly detects novelty when patterns don't match, learns actively by forming new generalizations.

The critical distinctions:

Causal structure versus semantic similarity: RAG might fail to match "glob" and "resonator" if text uses different terminology. Intuition matches the pattern "entity contact triggers property transfer" regardless of specific entity names.

Learning versus retrieval: RAG returns what was stored. Intuition consolidation generates generalizations not present in any single episode.

Novelty detection: RAG always retrieves something (best available match even if poor). Intuition explicitly flags when no patterns apply, triggering different processing.

These differences suggest Intuition and RAG serve different purposes. RAG augments models with static knowledge (documentation, references). Intuition enables continuous learning through experience accumulation and pattern extraction.

5.3 Relationship to Memory-Augmented Networks

Memory-augmented architectures like Neural Turing Machines (Graves et al., 2014) and Differentiable Neural Computers (Graves et al., 2016) provide networks with external memory and learned attention for reading/writing.

Similarities: Both separate storage from processing. Both use attention for retrieval. Both enable learning beyond parameter storage.

Differences:

Training: Memory networks train end-to-end through gradient descent. Episodic compression leverages pre-trained models without training.

Representation: Memory networks use learned distributed representations. Episodic compression uses symbolic natural language representations.

Interpretability: Memory networks produce opaque learned representations. Episodic compression produces human-readable generalizations with explicit causal structure.

Update mechanism: Memory networks use differentiable writes during training. Episodic compression uses consolidation through reasoning.

These architectural differences reflect different goals. Memory-augmented networks optimize task performance through representation learning. Episodic compression prioritizes interpretability and leveraging existing capabilities.

5.4 Relationship to Meta-Learning

Meta-learning approaches like MAML (Finn et al., 2017) enable rapid adaptation through initialization optimization.

Similarities: Both address sample-efficient learning. Both aim for few-shot adaptation. Both leverage prior experience.

Differences:

Adaptation: Meta-learning fine-tunes parameters on few examples. Episodic compression accumulates episodes and extracts generalizations without parameter updates.

Prior knowledge: Meta-learning encodes knowledge in optimized initialization. Episodic compression encodes knowledge in explicit generalizations.

Deployment: Meta-learning requires meta-training across task distributions. Episodic compression operates immediately with pre-trained models.

Interpretability: Meta-learning weight changes are opaque. Episodic compression generalizations are explicit and inspectable.

Meta-learning appears optimal when task distributions are known and meta-training data is available. Episodic compression may suit open-ended environments where task distributions are unknown or shifting.

6. Limitations and Future Work

6.1 Current Limitations

Empirical validation absent: This work presents architectural design without comprehensive experimental validation. Part 1 validated episodic compression at toy scale. This work proposes how that mechanism might scale but does not demonstrate the two-layer architecture operating with thousands of generalizations over extended periods. The proposals remain speculative pending systematic experimentation.

Optimal parameters unknown: Salience thresholds, consolidation frequency, generalization formation thresholds, context allocation strategies, and model size trade-offs all require empirical tuning. Proposed values (3-5 episode threshold, 5-10 episode consolidation batches, 2-6 hour fallback timers) represent educated guesses rather than validated parameters.

Threshold transitions uncertain: We estimated architecture becomes valuable around 50-100 generalizations with standard contexts or 200-400 generalizations with extended contexts. These estimates rest on token capacity calculations and qualitative factors. Actual transitions may occur

earlier, later, more gradually, or not at all. Usage patterns, domain complexity, and model capabilities likely affect transitions significantly.

Coherence at scale unproven: Maintaining consistency across thousands of generalizations may require more sophisticated mechanisms than proposed. Automatic contradiction detection might miss subtle inconsistencies. Resolution strategies for contradictory evidence need development and testing.

Long-term stability unknown: System behavior over months or years remains untested. Generalization quality might degrade. Consolidation times might grow prohibitively. Context management strategies might fail to maintain coherence. We lack evidence for sustained operation.

Cost projections uncertain: Operational cost estimates depend on actual usage patterns. If Executive engagement frequency exceeds projections or consolidation proves more expensive than anticipated, costs could substantially exceed estimates. Real-world deployment is needed to validate economic viability.

Alternative approaches unexplored: We propose one architectural design addressing scaling challenges. Alternative approaches (hierarchical RAG, progressive fine-tuning, hybrid systems) might address the same challenges differently or more effectively. Comparative evaluation awaits.

6.2 Open Questions

Consolidation timing: Does frequent micro-sleep suffice or do periodic longer consolidation cycles provide benefits? Human sleep serves functions beyond memory consolidation (cellular repair, metabolic regulation). Might analogous "deep consolidation" benefit artificial systems? Or does this reflect biological constraints irrelevant to computational systems?

Threshold mechanisms: Does the 3-5 episode threshold for generalization formation hold across domains? Does it vary by complexity or abstraction level? Can systems learn their own thresholds through meta-learning about when their generalizations prove reliable?

Forgetting strategies: What determines which episodes to retain versus prune after contributing to generalizations? Should forgetting be active (explicit deletion) or passive (decay over time)? How to balance generalization coverage with memory efficiency?

Meta-generalizations: Can systems develop generalizations about their own learning processes? Recognizing "my predictions about X domain are often wrong" or "I need more episodes before generalizing about Y" could enable adaptive learning strategies. Does this emerge naturally or require specific support?

Transfer between domains: How do generalizations from one domain interact with learning in related domains? Does physical causation knowledge transfer to social causation? Can systems identify structural similarities across domains?

Multi-agent learning: How should multiple systems with separate Intuition stores but shared experiences communicate learnings? Can one system's generalizations bootstrap another's? How to handle conflicting world models between agents?

Adversarial robustness: Can attackers form incorrect generalizations through strategic episode presentation? Do salience thresholds protect against spurious patterns? How quickly can systems recover from acquired false beliefs?

6.3 Future Research Directions

Real-world deployment study: Deploy two-layer architecture across diverse users and domains for extended periods (months). Measure generalization accumulation rates, prediction accuracy over time, consolidation efficiency and timing, cost per interaction as knowledge scales, and user satisfaction. Compare against baselines (static models, simple RAG, fine-tuning approaches, single-layer implementations).

Threshold validation: Systematically vary generalization formation thresholds (2-8 episodes) measuring false pattern rates versus missed pattern rates. Identify optimal thresholds for different domain complexities. Investigate whether systems can learn appropriate thresholds through meta-learning.

Parameter optimization: Develop methods for systems to optimize their own parameters: learning appropriate salience thresholds from prediction accuracy, adjusting consolidation frequency based on episode patterns, tuning formation thresholds based on false pattern rates.

Active learning extension: Beyond passive salience detection, develop curiosity-driven exploration: systems identifying high-value experiences to pursue, requesting specific information to resolve ambiguities, designing "experiments" to test uncertain generalizations.

Explanation generation: Develop transparent prediction mechanisms: "Based on generalization G7 (supported by episodes E3, E12, E44), I predict..." Enable users to challenge or correct specific knowledge. Build trust through transparent reasoning.

Adversarial testing: Test resilience against deliberately misleading experiences. Can strategic episode presentation form incorrect generalizations? Do salience thresholds protect adequately? How quickly do systems recover from false beliefs?

Hierarchical generalization: Current framework treats generalizations as flat. Investigate meta-generalizations (patterns about patterns), domain clustering (grouping related generalizations), abstraction levels (specific patterns versus general principles).

Multi-modal extension: Extend beyond language to visual experiences (images, video), auditory experiences (speech, environmental sounds), and multimodal experiences requiring cross-modal generalization.

7. Conclusion

Part 1 demonstrated that episodic compression may enable sample-efficient learning through experimental validation in an artificial world. This work addresses how that mechanism might scale beyond toy demonstrations to real-world continuous learning systems.

The critical observation is that episodic compression architecture provides value specifically at scale when world models exceed single-context capacity. At toy scales – dozens of generalizations fitting comfortably within standard contexts – specialized machinery provides no measurable advantage over single-model approaches. The architecture's benefits emerge precisely when accumulated knowledge exceeds what single contexts handle effectively.

We propose a two-layer design separating fast pattern-matching (Intuition) from deliberate reasoning (Executive). This separation appears analogous to distinctions in cognitive neuroscience between automatic and controlled processing systems (Schneider & Shiffrin, 1977; Kahneman, 2011), though our computational implementations differ from biological substrates. Intuition maintains consolidated knowledge and performs pattern-matching through attention mechanisms across large contexts using small efficient models. Executive conducts reasoning and learning through periodic consolidation cycles using large capable models within standard contexts.

This design potentially addresses scaling challenges: context limitations through separated storage and processing, retrieval efficiency through small model pattern-matching, real-time constraints through parallel processing and background consolidation, coherence through consistency checking, and computational costs through selective engagement of expensive processing.

The architecture distinguishes itself from retrieval-augmented generation through several mechanisms: pattern-matching on causal structure rather than semantic similarity, active learning through consolidation rather than static retrieval, and explicit novelty detection rather than best-match retrieval. These differences suggest complementary purposes – RAG augments with static knowledge while episodic compression enables continuous learning.

Implementation leverages existing pre-trained language models through architectural design rather than requiring new training methods. Small models with large contexts serve as Intuition. Large capable models serve as Executive. Structured prompting implements consolidation through inference rather than gradient descent. This approach offers potential practical advantages: immediate deployment, accessible costs through selective processing, and interpretable knowledge representations.

However, we emphasize several important limitations. This work remains conceptual – comprehensive empirical validation at scale awaits future research. Optimal parameters require empirical tuning. Threshold transitions remain uncertain. Long-term stability is unproven. Cost projections rest on assumptions about usage patterns. Alternative approaches addressing the same challenges may prove equally or more effective.

The relationship to LeCun's JEPA framework appears complementary rather than contradictory. Both address sample-efficient learning through better representations. JEPA emphasizes learned representations through gradient descent on embodied experiences. Episodic compression emphasizes natural compression through salience with narrative bootstrapping. Embodied learning may prove optimal for physical manipulation tasks. Episodic compression may prove optimal for abstract reasoning and social intelligence. Both contribute to understanding paths toward general intelligence.

This work extends Part 1's experimental validation with architectural proposals for scaling continuous learning. The two-layer design suggests how episodic compression might scale beyond controlled demonstrations toward practical deployment. Future work must validate this architecture through extended real-world deployment, systematic parameter optimization, and comparative evaluation against alternatives.

References

Core Framework

1. LeCun, Y. (2022). A path towards autonomous machine intelligence. Open Review, 62.
2. Mossrake Group, LLC. (2025a). Learning Through Narratives: Episodic Compression to Latent Variables for Sample-Efficient Intelligence. Part 1 of a series.

Episodic Memory and Learning

3. Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson (Eds.), Organization of memory (pp. 381-403). Academic Press.
4. Schacter, D. L., & Addis, D. R. (2007). The cognitive neuroscience of constructive memory: Remembering the past and imagining the future. Philosophical Transactions of the Royal Society B, 362(1481), 773-786.
5. Conway, M. A. (2009). Episodic memories. Neuropsychologia, 47(11), 2305-2313.
6. Ranganath, C., & Hsieh, L.-T. (2016). The hippocampus in time and space: Functional properties of episodic memory. Neuron, 90(2), 328–339.

Narrative Cognition

7. Bruner, J. (1986). Actual minds, possible worlds. Harvard University Press.
8. Bruner, J. (1990). Acts of meaning. Harvard University Press.
9. McAdams, D. P. (1993). The stories we live by: Personal myths and the making of the self. William Morrow.
10. Zacks, J. M., Speer, N. K., Swallow, K. M., Braver, T. S., & Reynolds, J. R. (2007). Event perception: A mind–brain perspective. Psychological Bulletin, 133(2), 273–293.
11. Kurby, C. A., & Zacks, J. M. (2008). Segmentation in narrative comprehension. Cognitive Psychology, 57(1), 25–61.

Curiosity and Intrinsic Motivation

12. Oudeyer, P. Y., & Kaplan, F. (2007). What is intrinsic motivation? A typology of computational approaches. Frontiers in Neurorobotics, 1, 6.
13. Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990-2010). IEEE Transactions on Autonomous Mental Development, 2(3), 230-247.

Sample Efficiency and Few-Shot Learning

14. Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. Science, 350(6266), 1332-1338.
15. Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. Behavioral and Brain Sciences, 40, e253.

Causal Learning

16. Pearl, J. (2009). Causality: Models, reasoning, and inference (2nd ed.). Cambridge University Press.
17. Gopnik, A., & Schulz, L. (2007). Causal learning: Psychology, philosophy, and computation. Oxford University Press.
18. Tenenbaum, J. B., Lake, B. M., Kemp, C., & Gershman, S. J. (2023). Compositional causal learning and time. Cognitive Science, 47(2), e13209.

Value Learning

19. Russell, S. (2019). Human compatible: Artificial intelligence and the problem of control. Viking.
20. Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in Neural Information Processing Systems, 30.

World Models

21. Ha, D., & Schmidhuber, J. (2018). World models. arXiv preprint arXiv:1803.10122.
22. Hafner, D., Lillicrap, T., Ba, J., & Norouzi, M. (2019). Dream to control: Learning behaviors by latent imagination. arXiv preprint arXiv:1912.01603.

Memory-Augmented Networks

23. Graves, A., Wayne, G., & Danihelka, I. (2014). Neural Turing machines. arXiv preprint arXiv:1410.5401.
24. Graves, A., Wayne, G., Reynolds, M., et al. (2016). Hybrid computing using a neural network with dynamic external memory. Nature, 538(7626), 471-476.

Meta-Learning

25. Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. International Conference on Machine Learning (ICML), 1126-1135.

Retrieval-Augmented Generation

26. Lewis, P., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.

Cognitive Systems

27. Kahneman, D. (2011). Thinking, fast and slow. Farrar, Straus and Giroux.
28. Schneider, W., & Shiffrin, R. M. (1977). Controlled and automatic human information processing: I. Detection, search, and attention. Psychological Review, 84(1), 1-66.
29. Shiffrin, R. M., & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending and a general theory. Psychological Review, 84(2), 127-190.

Catastrophic Forgetting

30. McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. Psychology of Learning and Motivation, 24, 109-165.
31. French, R. M. (1999). Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences, 3(4), 128-135.

Continual Learning

32. Kirkpatrick, J., et al. (2017). Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences, 114(13), 3521-3526.
33. Rusu, A. A., et al. (2016). Progressive neural networks. arXiv preprint arXiv:1606.04671.

Cognitive Architecture

34. Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). IEEE Transactions on Autonomous Mental Development, 2(3), 230–247.
35. Ha, D., & Schmidhuber, J. (2018). World Models. arXiv:1803.10122.
36. Hafner, D., Lillicrap, T., Ba, J., Fischer, I., & van den Oord, A. (2019). Learning latent dynamics for planning from pixels. Proceedings of ICLR.
37. Graves, A., Wayne, G., & Danihelka, I. (2014). Neural Turing Machines. arXiv:1410.5401.
38. Graves, A., Wayne, G., Reynolds, M., Harley, T., et al. (2016). Hybrid computing using a neural network with dynamic external memory. Nature, 538, 471–476.

Memory and Learning

39. Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson (Eds.), Organization of Memory (pp. 381–403). Academic Press.
40. Conway, M. A. (2009). Episodic memories. Neuropsychologia, 47(11), 2305–2313.
41. Schacter, D. L., & Addis, D. R. (2007). Constructive memory: The ghosts of past and future. Nature, 445(7123), 27–30.
42. Ranganath, C., & Hsieh, L.-T. (2016). The hippocampus in time and space: Functional properties of episodic memory. Neuron, 90(2), 328–339.

Temporal Modeling and Multi-Timescale Learning

43. Dai, Z., Wang, Y., & Dong, L. (2024). Transformer architectures for event time modeling. Proceedings of ICLR 2024.
44. Ahuja, A., & Singh, A. (2023). Memory-augmented transformers for continual learning. Proceedings of ICML 2023.
45. Kazemi, S. M., Goel, R., Kipf, T., Kazemi, A., Brubaker, M., & Hamilton, W. L. (2023). Temporal knowledge graphs: A survey. Journal of Artificial Intelligence Research, 77, 1335–1385.
46. Botvinick, M., Wang, J. X., Dabney, W., Miller, K. J., & Kurth-Nelson, Z. (2019). Reinforcement learning, fast and slow. Trends in Cognitive Sciences, 23(5), 408–422.
47. Kiebel, S. J., Daunizeau, J., & Friston, K. J. (2008). A hierarchy of time-scales in the brain. PLoS Computational Biology, 4(11), e1000209.
48. Tallec, C., & Ollivier, Y. (2018). Can recurrent neural networks warp time? Proceedings of ICLR 2018.

Version 1.0