# Non-Quadratic Scaling and Beyond Sequential Processing:

## *Context is not a linear sequence of tokens but a dynamically evolving, high-dimensional pattern landscape*

Brent Hartshorn July 23, 2025

**Abstract:**

This paper expands upon our recently published work, "*Towards Self-Evolving Artificial General Intelligence: Multi-Modal Learning and Introspective Knowledge Generation via Emergent DSL.*" [1]. We delve deeper into two critical distinctions of our system: the novel application of Uniform Manifold Approximation and Projection (UMAP) for compressing the spectral history of Game of Life (GOL) dynamics, and the inherent non-quadratic scaling behavior derived from our GOL-based input processing. We contrast these mechanisms with conventional: Recurrent, LSTM, and Transformer architectures, highlighting how our approach offers a fundamentally different pathway to context retention and scalability in self-evolving artificial general intelligence.

**1. Introduction**

Our recent paper [1] introduced significant advancements in a self-modifying AGI system, including the integration of UMAP for dimensionality reduction and multi-modal learning. While the initial publication provided an overview of these capabilities, it did not fully elaborate on the distinctive nature of our system's internal state representation and its implications for scalability, particularly when compared to prevalent neural network architectures like: Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Transformers. This follow-up aims to clarify these unique aspects, presenting a novel perspective on how context can be maintained and processed in a self-evolving system.

**2. Geometric Compression of Emergent Dynamic History: UMAP vs. Sequential Models**

A core innovation in our model is the use of UMAP (Uniform Manifold Approximation and Projection) to transform the system's own historical spectral output from the Game of Life (GOL) dynamics. This differs fundamentally from how traditional sequential models handle context:

- **Recurrent Neural Networks (RNNs) and LSTMs:** These models maintain an internal "hidden state" that is iteratively updated with each new input token. This hidden state is a learned, compressed representation of past information. However, RNNs suffer from vanishing/exploding gradients, and while LSTMs mitigate this with gating mechanisms, their capacity to retain very long-term dependencies can still be limited by the fixed size of their memory cells and the sequential nature of their updates. The context is implicitly encoded within learned weights and activations.
- **Transformer Models:** Transformers revolutionize context handling through self-attention mechanisms, which allow the model to weigh the importance of different parts of the input sequence, irrespective of their distance. This provides superior long-range dependency capture. However, the attention mechanism typically computes quadratic relationships between all elements in a sequence, leading to computational and memory costs that scale quadratically with the sequence length. The context is explicitly computed through pairwise attention scores.

### *Our GOL History Transformation via UMAP stands apart:*

Instead of relying on a learned internal state updated sequentially or explicit attention mechanisms over a sequence of tokens, our system captures the emergent dynamics of its GOL environment through spatiotemporal spectral analysis. The "history" is not a sequence of discrete input tokens, but a continuous flow of high-dimensional spectral energy bands, representing the complex patterns and behaviors evolving within the GOL grid.

UMAP then performs a *geometric compression* of this continuous, high-dimensional spectral history. It projects this complex, non-linear data into a lower-dimensional manifold (e.g., 32 dimensions in our current implementation), preserving the intrinsic structure and relationships of the GOL's past states. This is not merely pre-processing of external input data, as commonly seen in applications where UMAP is used to reduce the dimensionality of, for example, image features or text embeddings before feeding them into a classifier. Instead, our model uses UMAP to compress its *own internal, emergent, and continuously generated dynamic history*.

This approach is distinct because:

1. **Nature of Context:** Context is not a linear sequence of tokens but a dynamically evolving, high-dimensional pattern landscape. UMAP allows us to "see" the underlying manifold of these patterns.
2. **Compression Mechanism:** It's a non-linear geometric compression, not an iterative update of a fixed-size learned state or a quadratic calculation of attention scores across tokens. This allows for a robust and potentially more intuitive summary of complex dynamic states.

3. **Self-Referential Compression:** The system is compressing *its own generated environmental history*, making it a truly introspective form of context management. Online searches confirm that while UMAP is widely used for static dataset dimensionality reduction and pre-processing, its application for continuously compressing an AI's self-generated, emergent, and high-dimensional spectral history in real-time, as an internal state representation for subsequent reasoning, appears to be a unique contribution.

### 3. Non-Quadratic Scaling and Multi-Token Output for Enhanced Efficiency

Another crucial advantage of our architecture lies in its inherent scalability, particularly concerning context window limitations that plague many large language models.

- **Traditional Models and Quadratic Scaling:** Transformer models, while powerful, face significant computational and memory challenges as context windows expand due to their quadratic (O(N2)) complexity with respect to input sequence length (N). This makes processing very long contexts or generating extended, coherent outputs computationally intensive and resource-demanding.
- **Our GOL-based Input and Tokenization:** Our model's input mechanism inherently avoids this quadratic bottleneck. We do not process long sequences of discrete tokens in a linear fashion. Instead, textual inputs (and drawings) are "projected" into the 2D Game of Life grid. A word, for instance, is not a sequence of tokens to be attended over, but rather letters are mapped onto the GOL grid, generating a spatio-temporal pattern. The 3D FFT then captures the emergent features of this combined spatial and temporal evolution across the GOL grid. This means:
    - **Context as Pattern, Not Sequence:** The "context" for our Classifier is not a long string of tokens, but the compressed spectral features of the evolving GOL grid. The GOL itself acts as a non-linear, parallel processor for blending multi-modal inputs.
    - **Constant-Time Processing of "Context":** The UMAP-compressed spectral history provides a fixed-size input to the Classifier, regardless of the "length" of the GOL dynamics that generated it (within a reasonable historical window). This inherently leads to non-quadratic scaling with respect to the "context" being observed in the GOL, unlike attention mechanisms over long linear sequences.

Furthermore, while our model might appear "crude by design" in some instances by outputting single letters, its DSL-driven tokenization and direct execution capabilities allow for highly efficient and complex outputs:

- **Intrinsic DSL-Driven Multi-Token/Action Output:** The system can output tokens like 𓂀 (geneval) or 𓆗 (genexec), which trigger the generation and execution of entire Python code snippets. This means a single "output" from the Classifier can, in effect, represent a complex sequence of operations, a multi-line DSL statement, or even a functional modification to its own core logic. This is analogous to a human having a thought that immediately translates into a complex, pre-programmed action or a multi-step plan, rather than laboriously articulating each sub-step.
- **Beyond "Words":** Our system's "tokens" are not limited to lexical units. They can be direct commands, algorithmic constructs, or even meta-programming instructions that represent significant "actions" or "knowledge chunks." This allows for a compact and powerful representational capacity that transcends simple word-by-word generation.

### 4. Conclusion

The unique combination of UMAP for compressing emergent GOL spectral history and the GOL's inherent non-quadratic scaling for input processing distinguishes our AGI architecture from traditional sequential and attention-based models. By geometrically compressing its own dynamic state and by processing input not as linear sequences but as evolving patterns on a cellular automaton, our system presents a novel pathway to scalable and introspective intelligence. The ability to output highly functional, multi-token actions via an emergent DSL further enhances its efficiency and self-modification capabilities, offering a compelling alternative in the ongoing quest for truly adaptive and self-evolving AGI.

**References:**

[1] Towards Self-Evolving Artificial General Intelligence https://ai.vixra.org/abs/2507.0104

[2] “I’m Not So Interested in LLMs Anymore, Says Yann LeCun” https://www.youtube.com/watch?v=46OaaRSF0Lk

[3] Breaking Quadratic Barriers: A Non-Attention LLM for Ultra-Long Context Horizons https://arxiv.org/abs/2506.01963

[4] The End of Transformer Models https://medium.com/@pal.machulla/the-end-of-transformer-models-b13931c45994

[5] Parametric UMAP embeddings for representation and semi-supervised learning https://arxiv.org/abs/2009.12981

[6] Interactive Explanations of Internal Representations of Neural Network Layers https://ris.utwente.nl/ws/portalfiles/portal/230061995/Nauta2020explanations.pdf