PAIRWISE PHASE SPACE EMBEDDING IN TRANSFORMER ARCHITECTURES

*

Kevin R. Haylett, PhD
Manchester, UK
kevin.haylett@gmail.com

ABSTRACT

The Transformer architecture's "attention" mechanism, heralded as a cornerstone of large language models, is misnamed, obscuring its true nature as a pairwise phase-space embedding rooted in nonlinear dynamical systems. This paper demonstrates that the dot-product similarity operations—termed "query," "key," and "value"—mirror delay-coordinate embedding techniques pioneered by Takens and others in the 1980s[1,2]. By comparing time-shifted token projections, Transformers reconstruct a latent language attractor, transforming sequential data into a high-dimensional manifold where meaning emerges as geometric trajectories, not cognitive focus. This re-framing, inspired by prior work in high-dimensional signal clustering, reveals that positional encodings and softmax normalization are often redundant, as temporal structure is inherently captured in delay-based geometries. This work points to retiring the term "attention" in favour of "pairwise phase space embedding," offering a clearer, finite, and interpretable framework aligned with Finite Mechanics principles—a framework privileging geometric constraints over infinite parameterization. This shift suggests leaner architectures, bypassing encodings and reducing computational complexity, while enhancing transparency to mitigate risks like manifold distortions. Grounded in historical parallels from neurophysiology, cardiology, and seismology, this reinterpretation positions the embedding mechanism more formally in non-linear dynamics. This framing shows how delay embeddings inherently encode positional information, possibly rendering softmax and positional encodings redundant. Transformers can be seen as an unknowing rediscovery of dynamical systems methods, opening paths to principled, geometry-driven models.

***K*eywords** Tranformer Architecture · Dynamical Systems · Delay Embeddings · Phase Space Embedding · Finite Mechanics · Neural Geometry

1 Introduction

The architecture commonly referred to as "attention" has become the cornerstone of modern large language models. It is described using terms such as "query," "key," and "value," which borrow language from human cognition and database systems, possibly giving an illusion of interpretive or selective focus. However, close inspection reveals that this mechanism is neither cognitive nor attentional in any meaningful sense. It is, at its core, a structured similarity operation between projected vectors, a dot product followed by normalization. What it does, mechanistically, is not "attend," but measure proximity in a latent space, a technique long understood in modern dynamical systems analysis. In the case of the LLM, it serves to convert a time series of tokens into a two-dimensional format suitable for presentation to a multi-perceptron neural network.

This paper therefore proposes that such a mechanism is more accurately and productively understood as a form of phase space embedding, a technique drawn from the study of nonlinear dynamical systems. Originally developed by Takens,

Citation: Kevin R. Haylett, "Pairwise Phase Space Embedding in Transformer Architectures," preprint (May 2025), available at https://finitemechanics.com/papers/phase-space-transformers.pdf. Draft version prepared for submission to arXiv [cs.LG]

Packard, and others in the 1980s, phase space embedding allows a one-dimensional time series to be reinterpreted as a multidimensional trajectory, revealing the hidden structure of the system that generated it. It is a method not of storing memory, but of reconstructing it spatially.

The similarity operation at the heart of so-called attention, pairwise dot products between shifted representations of the same sequence, performs this same function. It constructs a surrogate space in which sequential information is preserved through relative positioning. Each token in a sequence is compared to every other, not to decide "what to attend to," but to reconstruct a geometry of meaning from which linguistic or semantic predictions can be made. What emerges is not a focus of attention, but a trajectory across an attractor manifold formed by language itself.

The purpose of this paper is to formalize that equivalence. We begin by outlining the theory of phase space embedding, tracing its origin in nonlinear science. We then demonstrate that transformer-based architectures perform a structurally equivalent operation, albeit one phrased in a language that suggests a semantic or interpretive process. We propose that reframing this operation as a nonlinear dynamical systems approach has practical consequences: it allows us to simplify components of the transformer, challenge the necessity of positional encodings, and potentially reduce computational complexity while improving interpretability.

By grounding modern neural sequence processing of LLM tokens in the formal and well-understood mathematics of dynamical systems, we open a path toward more principled, finite, and explainable models, of which the transformer is only a special, unknowing case.

2 Phase Space Embedding Theory

2.1 Origin in Nonlinear Dynamics

In the 1970s and 1980s, a new approach to analyzing complex systems began to take form across disciplines such as cardiology, meteorology, and fluid dynamics. Systems that were previously seen as chaotic or unpredictable were now being modeled not by linear differential equations, but through reconstruction of their underlying geometry. This was the birth of modern nonlinear dynamical systems theory, and one of its most profound contributions was the technique known as phase space embedding.

Pioneered by Floris Takens[1], James P. Crutchfield[5], Robert Shaw[6], and later expanded by Leon Glass and others[3], phase space embedding provided a method to reconstruct the state space of a dynamical system from a single observable time series. In simple terms, this meant that even if we could only measure one aspect of a system, we could still recover the system's internal structure and dynamics.

The key to this process was the method of delays. By recording not just the current measurement, but also its values at previous time steps, one could construct a trajectory in a higher-dimensional space. This trajectory unfolds the latent attractor that governs the system's evolution. What initially appears as a flat or noisy signal becomes a geometric object, a path through a structured manifold in phase space.

2.2 Embedding a Time Series

Mathematically, delay embedding works by mapping a one-dimensional sequence into an n-dimensional space through time-shifted copies of itself. Given a time series x(t), we construct vectors of the form:

x(t) = [x(t), x(t - tau), x(t - 2tau), ..., x(t - (m - 1)tau)]

Here, m is the embedding dimension, and tau is the delay. Takens' theorem guarantees that if m is sufficiently large, the resulting reconstruction is a diffeomorphic image of the original attractor, meaning it preserves the system's qualitative behavior and structure. A diffeomorphic image is a smooth, reversible mapping that preserves the attractor's geometric structure, ensuring the embedded trajectory reflects the system's dynamics, such as loops or convergence patterns.

The effect is striking. What was once a linear or one-dimensional sequence is now a trajectory in space, whose geometry can be analyzed, visualized, and used for prediction or classification. This approach has been used to analyze heartbeat dynamics, atmospheric data, and stock market patterns. It is also at the heart of manifold learning methods used in many machine learning algorithms today.

Crucially, this embedding process does not add information. It simply re-represents the existing time series in a way that reveals its underlying structure. It is a transformation, not a translation. This is what makes it so powerful: it exposes hidden order within apparent complexity.

2.3 A Language Example: Sentence as Time Series

To make the connection between time series embedding and language more explicit, consider the simple case of a sentence treated as a discrete sequence of tokens. Each word in a sentence occurs in a fixed order, and that order imparts structure. From a dynamical systems perspective, this is a form of temporal evolution. Each word corresponds to a distinct point in time, and the sentence as a whole is a time series of symbolic or numerical data. In this context, the language attractor is the latent manifold of semantic and syntactic relationships among tokens. Delay embedding reconstructs this attractor as a geometric trajectory, per Takens' theorem, encoding the sentence's meaning in its shape.

Let us take a simple sentence:

"The quick brown fox jumps over the lazy dog happily today before tea."

We can map each word to a number, using a stand-in for a learned embedding. For illustration purposes, we use word length as a proxy:

[3, 5, 5, 3, 5, 4, 3, 4, 8, 5, 5, 6, 3]

This one-dimensional series represents our time signal. We now apply the method of delays to embed this series into a two-dimensional space using Takens' approach. Using an embedding dimension of 2 and a delay tau = 1, we construct the following vectors:

x1 = [3, 5]

x2 = [5, 5]

x3 = [5, 3]

x4 = [3, 5]

x5 = [5, 4]

...

Each vector represents a point in 2D space. Plotting these sequentially produces a visible trajectory, a path, through this new phase space. What was previously a linear signal now reveals turning points, recurrences, and geometrical structure. This is the core insight of phase space embedding: meaning is not stored in the values themselves, but in the shape they collectively form over time.

Transformer architectures perform an analogous operation, although this is not typically acknowledged. By computing dot products between token projections, they effectively measure geometric relationships between word embeddings that are shifted versions of the same sentence. The result is a high-dimensional manifold that encodes the sentence not as a list of words, but as a spatial configuration, a trajectory of relationships. This latent space is what enables prediction, coherence, and contextual adaptation.

In both cases, a linear sequence is transformed into a structured path. The phase space view provides a clean and unambiguous way of understanding this transformation, without relying on metaphors such as attention or focus. It also opens the door to visualizing and interpreting the language manifold as a dynamic geometry, rather than a table of weights.

3 Application to Transformer and Neural Architectures

3.1 Mechanistic Breakdown of the Transformer

The Transformer, introduced by Vaswani *et al.* [4], revolutionized neural language models by replacing recurrent structures with a feedforward pipeline, enabling parallelism and unprecedented scalability. Its core mechanism, misleadingly termed "attention," relies on algebraic operations described with anthropomorphic labels: "query," "key," and "value." These terms suggest cognitive or information-retrieval processes, but the reality is purely computational.

For a sequence of n tokens, each represented by an embedding vector $\mathbf{e}_i \in \mathbb{R}^d$, the Transformer computes three projections per token:

$$\mathbf{q}_i = W_Q \mathbf{e}_i, \quad \mathbf{k}_i = W_K \mathbf{e}_i, \quad \mathbf{v}_i = W_V \mathbf{e}_i, \tag{1}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ are learned linear transformation matrices, and $\mathbf{q}_i, \mathbf{k}_i, \mathbf{v}_i \in \mathbb{R}^d$ are the query, key, and value vectors, respectively. Contextual similarity is computed via the dot product between each query and every key,

forming a similarity matrix $A \in \mathbb{R}^{n \times n}$:

$$A_{ij} = \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d}}. \tag{2}$$

The scaling factor $\sqrt{d}$ prevents exploding gradients. This matrix is normalized using a softmax function to produce weights:

$$W_{ij} = \text{softmax}(A_i)_j = \frac{\exp(A_{ij})}{\sum_{k=1}^{n} \exp(A_{ik})}. \tag{3}$$

These weights are applied to the value vectors to compute a new representation for each token:

$$\mathbf{c}_i = \sum_{j=1}^{n} W_{ij} \mathbf{v}_j. \tag{4}$$

This process, termed "scaled dot-product attention," is repeated across multiple heads and stacked in layers, with feedforward networks interleaved. Positional encodings—vectors added to embeddings to encode token order—and optional masking (e.g., zeroing future tokens in autoregressive models) ensure sequential coherence.

Far from cognitive "attention," this is a pairwise similarity measurement across a sequence, transforming a temporal series into a weighted spatial configuration. It constructs a latent geometry, not a focus of intent.

3.2 Demonstrating the Embedding Equivalence

Viewing the Transformer through the lens of nonlinear dynamical systems reveals a striking equivalence to phase-space embedding. Consider a sequence of tokens $\{t_1, t_2, \ldots, t_n\}$ as a discrete time series, where each token t_i is embedded as $\mathbf{e}_i \in \mathbb{R}^d$. The Transformer's dot-product operation compares projections of these embeddings, effectively measuring relationships between time-shifted representations of the sequence.

In phase-space embedding, a time series $x(t)$ is mapped to a higher-dimensional space using delay coordinates:

$$\mathbf{x}(t) = [x(t),\, x(t-\tau),\, x(t-2\tau),\, \ldots,\, x\,(t-(m-1)\tau)] \tag{5}$$

where m is the embedding dimension and τ is the delay. Takens' theorem ensures that, for sufficient m, this reconstruction preserves the system's attractor geometry. The Transformer performs a structurally similar operation. The similarity matrix $A_{ij} = \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d}}$ quantifies the geometric proximity between token i's query and token j's key, akin to comparing delayed vectors in a phase-space trajectory.

Formally, let $\mathbf{e}_i$ represent the state of the sequence at time i. The query and key projections ($\mathbf{q}_i = W_Q \mathbf{e}_i, \mathbf{k}_j = W_K \mathbf{e}_j$) are analogous to time-shifted coordinates, as W_Q and W_K apply different transformations to the same underlying embeddings. The dot product $\mathbf{q}_i \cdot \mathbf{k}_j$ measures their alignment, constructing a surrogate space where temporal relationships are encoded as spatial distances. The weighted sum $\mathbf{c}_i = \sum_j W_{ij} \mathbf{v}_j$ then blends these relationships into a new representation, unfolding the sequence's latent manifold layer by layer.

To formalize the equivalence between Transformer operations and phase space embedding, consider a sequence of tokens

$$\{t_1, t_2, \ldots, t_n\},$$

each embedded as

$$\mathbf{e}_i \in \mathbb{R}^d.$$

The Transformer computes query and key vectors as

$$\mathbf{q}_i = W_Q \mathbf{e}_i, \quad \mathbf{k}_j = W_K \mathbf{e}_j,$$

where

$$W_Q, W_K \in \mathbb{R}^{d \times d}$$

are linear transformations. The scaled dot product

$$A_{ij} = \frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d}}$$

measures geometric alignment between these projections, analogous to comparing delay vectors in phase space:

$$\mathbf{x}(t_i) = [\mathbf{e}_i, \mathbf{e}_{i-1}, \ldots, \mathbf{e}_{i-m+1}], \quad \mathbf{x}(t_j) = [\mathbf{e}_j, \mathbf{e}_{j-1}, \ldots, \mathbf{e}_{j-m+1}],$$

where

$$\mathbf{q}_i \cdot \mathbf{k}_j \sim \langle \mathbf{x}(t_i), \mathbf{x}(t_j) \rangle$$

for a similarity measure $\langle \cdot, \cdot \rangle$ (e.g., inner product).

Per Takens' theorem, if the embedding dimension d is sufficiently large, this pairwise comparison reconstructs a diffeomorphic image of the language attractor—a high-dimensional manifold encoding the sequence's semantic and syntactic structure. Thus, the similarity matrix

$$A \in \mathbb{R}^{n \times n}$$

represents a trajectory through this latent space, unfolding the temporal sequence into a geometric configuration without requiring explicit normalization or positional markers.

This is not "attention" but a reconstruction of a language attractor. Each Transformer layer refines this geometry, embedding the sequence into increasingly structured contexts, much like successive delay embeddings unfold a dynamical system's trajectory.

To illustrate, revisit the sentence "The quick brown fox jumps over the lazy dog happily today before tea" from Section 2.3, with word-length embeddings [3,5,5,3,5,4,3,4,8,5,5,6,3]. In a Transformer, these tokens are projected into $\mathbf{q}_i, \mathbf{k}_i, \mathbf{v}_i$, and the similarity matrix A captures pairwise relationships (e.g., "quick" aligns with "brown" due to syntactic proximity). This matrix mirrors the 2D trajectory ([3,5], [5,5], ...) formed by delay embedding, where geometric structure encodes sequential meaning. The Transformer's output is a path through a high-dimensional manifold, not a selection of "attended" tokens.

3.3 Simplification Opportunity

Recognizing the Transformer as a phase-space embedding opens avenues for simplification. In traditional delay embedding, temporal information is inherent in the relative placement of delay vectors—no explicit positional encodings are needed. The Transformer's reliance on positional encodings, added to embeddings to preserve order, may be redundant if delay-style relationships are directly leveraged. For instance, instead of adding sinusoidal or learned positional vectors, the sequence could be embedded as:

$$\mathbf{x}_i = [\mathbf{e}_i, \mathbf{e}_{i-1}, \ldots, \mathbf{e}_{i-m+1}], \tag{6}$$

where past tokens form a delay coordinate, capturing temporal structure geometrically. This is aligns with Takens' theorem where delay embeddings capture temporal order through the relative positioning of vectors, which reconstructs the sequence's attractor geometry without external markers. For example, in the sequence

$$[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3],$$

the delay vectors

$$[\mathbf{e}_1, \mathbf{e}_2], \quad [\mathbf{e}_2, \mathbf{e}_3]$$

encode order inherently, rendering the Transformer's sinusoidal positional encodings redundant.

Moreover, softmax normalization and masking (e.g., zeroing future tokens in autoregressive models) are corrective measures to stabilize a process not understood as delay embedding. Per Takens' theorem, the attractor's geometry constrains relationships, rendering softmax unnecessary, as simpler metrics like cosine similarity can preserve the manifold's structure.

Unlike softmax, which normalizes dot products to stabilize training, delay embeddings rely on the attractor's intrinsic geometry to constrain relationships. Per Takens' theorem, the manifold's structure preserves the system's dynamics without such corrections, suggesting simpler metrics like cosine similarity can suffice.

The Transformer's softmax normalization, while critical for stabilizing gradient updates in variable-length sequences, is unnecessary in delay embeddings. Takens' theorem ensures that temporal structure is preserved by the attractor's geometry, not by normalized weights. For instance, the delay vectors $[e_i, e_{i-1}]$ and $[e_j, e_{j-1}]$ encode order inherently through their relative positions in phase space. This suggests that softmax—introduced to manage unbounded dot products in long sequences—can be replaced with fixed-scale similarity measures once the manifold's structure is explicitly leveraged.

Softmax normalization in Transformers compensates for unbounded dot products in variable-length sequences—a problem absent in delay embeddings, where the attractor's geometry intrinsically bounds pairwise relationships. This suggests softmax is a computational crutch, not a theoretical necessity. While softmax aids gradient stability in practice, its role diminishes if embeddings explicitly reconstruct the language attractor's topology.

These simplifications suggest a leaner architecture: one that embeds sequences directly via delay coordinates, bypasses positional encodings, and uses geometric constraints for contextual blending. A preliminary experiment could test

this by comparing a shallow model with delay-embedded tokens to a standard Transformer, measuring perplexity and efficiency. Such a design would be more interpretable, computationally lighter, and aligned with the finite, geometric principles of Finite Mechanics.

3.4 Implications for a Simpler Approach

The Transformer's *attention* mechanism was originally a pragmatic engineering solution: it converted serial token sequences into a 2D similarity matrix for parallel computation. Positional encodings and softmax normalization were *ad hoc* additions to preserve order and stabilize training—unaware that the method of delays already inherently encodes temporal structure through phase-space geometry.

In fact, we could construct an equivalent square matrix for parallel processing directly from delay embeddings: by stacking delay vectors (e.g., $x_i = [e_i, e_{i-1}, \dots]$) as rows or columns, padding as needed. This would eliminate the need for positional encodings and softmax, as the attractor's geometry naturally bounds relationships. The Transformer, unknowingly, reinvented dynamical embedding—but with redundant corrections.

4 Historical Parallels in Signal Analysis

Before neural networks came to dominate machine learning, a wide range of problems in medicine, physics, and engineering were addressed using techniques from nonlinear dynamical systems. These approaches often relied on time series data, raw sequences of measurements that appeared noisy or complex at first glance, but which revealed deep structure when reinterpreted in geometric terms.

Among the earliest and most successful applications of phase space embedding was the analysis of biological rhythms. Leon Glass and Michael Mackey applied these techniques to understand cardiac dynamics, particularly arrhythmias and heart rate variability. In their work, electrocardiogram signals were not treated as isolated peaks and troughs, but as trajectories within a latent physiological state space. Delay embedding allowed researchers to visualize how the heart's electrical behavior evolved over time, detecting emergent patterns, limit cycles, or chaos.

Similar strategies were used in the study of neurological data. Electroencephalogram recordings were reanalyzed using delay coordinates, uncovering signatures of epilepsy, sleep stages, and even cognitive attention as geometric phenomena rather than statistical events. These embeddings helped classify states not by fixed thresholds, but by their trajectories within a reconstructed attractor space.

In seismology, time-delay embeddings were employed to detect precursors to earthquakes. In audio processing, similar embeddings were used to distinguish between phonemes, speaker identities, and emotional tone, by embedding waveform snippets into geometric manifolds.

What unites these applications is a shift in focus: from statistical averaging to structure reconstruction. Delay embedding transforms a time series into a map of the system that generated it, allowing for richer analysis without needing to observe every internal variable directly. This approach does not rely on massive parameterization or deep models, it leverages the intrinsic structure already present in the data.

In many ways, the operations at the heart of transformer architectures are closer to these earlier dynamical techniques than to traditional feedforward neural networks. However, this lineage has gone largely unacknowledged. The conceptual heritage of Takens, Packard, and Glass is absent from the vocabulary of deep learning. The emphasis on scaling, stacking, and parameter tuning has obscured the fact that the fundamental operation of pairwise similarity across time is a known and well-theorized method for reconstructing dynamical systems.

Recognizing this parallel provides not just historical grounding, but an opportunity. It suggests that we can revisit the transformer not as a singular invention, but as a rediscovery, one that might benefit from reconnecting with its true intellectual ancestry.

5 Discussion

The recognition that transformer architectures are performing a form of phase space embedding, rather than "attention," reframes a significant portion of modern machine learning. It removes the cognitive metaphor that has dominated discourse and replaces it with a geometric and mechanical interpretation, rooted in nonlinear dynamical systems.

This reframing carries several implications:

5.1 Terminological Clarity

The language of "attention," while rhetorically effective, has introduced persistent confusion. It implies intentionality, selection, or interpretive focus, none of which are present in the actual operation. As this paper has shown, what is being computed is a structural similarity between projections of the same system across time. This is not attention, but trajectory reconstruction.

By naming this mechanism more accurately, as pairwise phase space embedding, we realign our understanding with the actual geometry of what is taking place, and avoid anthropomorphizing processes that are neither cognitive nor semantic in nature.

5.2 Architectural Consequences

Recognizing the transformer as a system for unfolding phase space leads naturally to reconsiderations of its design. In traditional delay embedding, no positional encoding is required; time is encoded in the structure of the vector itself. Likewise, masking and normalization techniques such as softmax can be understood as corrective overlays introduced to stabilize a process whose geometric nature was not fully recognized.

The reliance on softmax reflects a misunderstanding of the underlying geometry. In delay embeddings, pairwise comparisons are inherently bounded by the attractor's topology, obviating the need for normalization. Future architectures could adopt manifold-constrained similarity metrics, bypassing softmax entirely.

Positional encodings simulate delay structure artificially (e.g., via sinusoidal waves), whereas delay embeddings *are* the structure. The latter is more parsimonious but may require careful tuning of m and τ.

This opens the door to simplified architectures that rely on delay-style embeddings directly, avoid unnecessary positional signals, and use geodesic or curvature-based metrics instead of matrix-based similarity. Such systems would be more efficient, more interpretable, and more finite, qualities aligned with the goals of Finite Mechanics.

5.3 Conceptual Consequences

Framing the language manifold as a dynamic attractor space, rather than a parameterized token map, supports an entirely different view of cognition and computation. Sentences are no longer generated token by token, but traced as paths across a learned manifold, guided by field structure rather than probabilistic sampling. This resonates strongly with field-based theories of meaning, language as motion, and interaction-based modeling.

It also challenges the default paradigm of neural language models as infinite statistical engines. Instead, it suggests a finite dynamic core: one that operates through geometric interaction and internal constraint, rather than brute-force function approximation.

5.4 Philosophical Alignment

This reinterpretation of transformer mechanics through the lens of phase space is not merely a technical substitution. It is a philosophical realignment. It returns us to a view of systems not as networks of weights and losses, but as fields of interaction unfolding in time. It privileges geometry over mystique, structure over metaphor.

In doing so, it makes models more explainable, more grounded, and more capable of integration into a broader scientific worldview, one that includes physiology, cognition, and semantics under the shared language of finite dynamics.

6 Conclusion

This paper has demonstrated that the mechanism popularly known as "attention" within transformer-based neural networks is more accurately described as a form of pairwise phase space embedding. By revisiting the origins of this technique in nonlinear dynamical systems, particularly through the work of Takens, Packard, and Glass, we have shown that the essential operation of the transformer is not cognitive, semantic, or attentional, it is geometrical. It constructs a latent attractor space from a time series through delay-structured pairwise comparisons.

We have further illustrated that this same mechanism has long been employed in fields such as cardiology, seismology, and signal processing, where it is explicitly recognized as a method of system reconstruction, not interpretation. The similarity operations within the transformer serve the same function, but have been described through an anthropomorphic vocabulary that has obscured both their origin and their potential.

Recognizing this equivalence enables a simplification of neural architecture design. By reframing these operations as geometric projections within a dynamical manifold, we open the door to models that are more explainable, more

efficient, and better aligned with the foundational principles of Finite Mechanics. Positional encodings, masking procedures, and softmax normalization may be re-evaluated in light of this insight and replaced by delay-embedding strategies that are formally grounded and computationally simpler.

This paper serves as the first in a two-part contribution. The companion work, to appear in Finite Tractus: Part II, will introduce a new dynamical architecture based on hyperspherical manifold geometry and magnetically interacting word identities. That model will extend the present analysis into a generative field system where language is not sampled but traced, and where sentences emerge as paths through a structured, charged semantic topology.

This reinterpretation is not a rebranding of the Transformer—it is a clarification of what it has been all along. What was once described as attention is better understood as dynamical embedding. The implications of this shift reach far beyond architecture, pointing toward a future in which intelligence is modeled not through abstraction, but through finite geometry, structure, and interaction.

References

[1] Takens, F. (1981). Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980* (pp. 366–381). Springer.

[2] Packard, N. H., Crutchfield, J. P., Farmer, J. D., & Shaw, R. S. (1980). Geometry from a time series. *Physical Review Letters*, 45(9), 712–716.

[3] Glass, L., & Mackey, M. C. (1988). *From Clocks to Chaos: The Rhythms of Life*. Princeton University Press.

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (pp. 5998–6008).

[5] Crutchfield, J. P., & Packard, N. H. (1982). Symbolic dynamics of noisy chaos. *Physica D: Nonlinear Phenomena*, 7(1–3), 201–223.

[6] Shaw, R. (1984). *The Dripping Faucet as a Model Chaotic System*. Aerial Press.

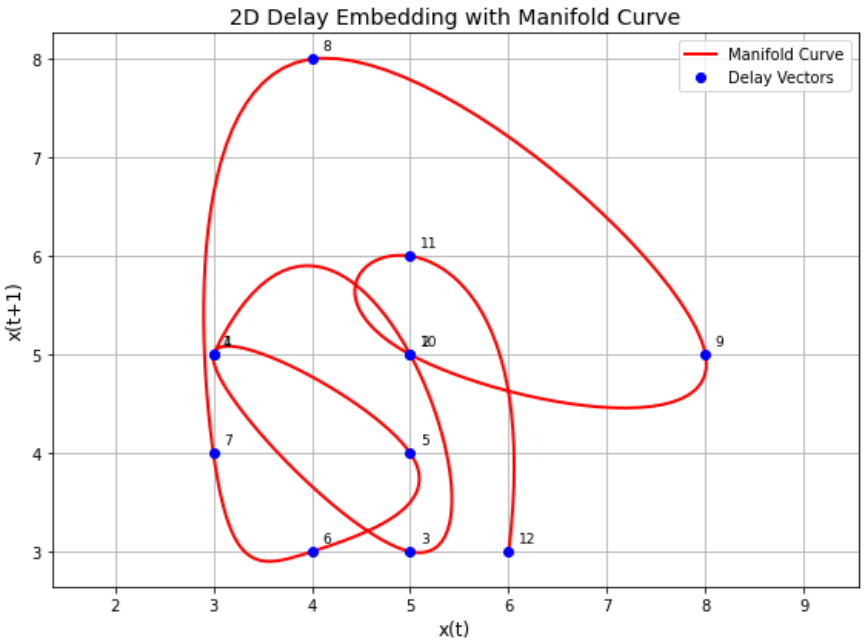

Figure 1: **2D Delay Embedding with Smooth Manifold Approximation.** The original word-length time series from a sentence is plotted as a set of delay vectors $(x(t), x(t+1))$, each representing a step through phase space. A smooth spline curve (red) suggests the latent manifold structure implicitly reconstructed by the delay embedding. This geometric trajectory illustrates how temporal patterns can be encoded without cognitive or attentional operations—supporting the reinterpretation of Transformer mechanics as dynamical embedding. Note how the trajectory's curvature encodes word-order relationships (e.g., 'quick'→'brown') without softmax or positional markers.

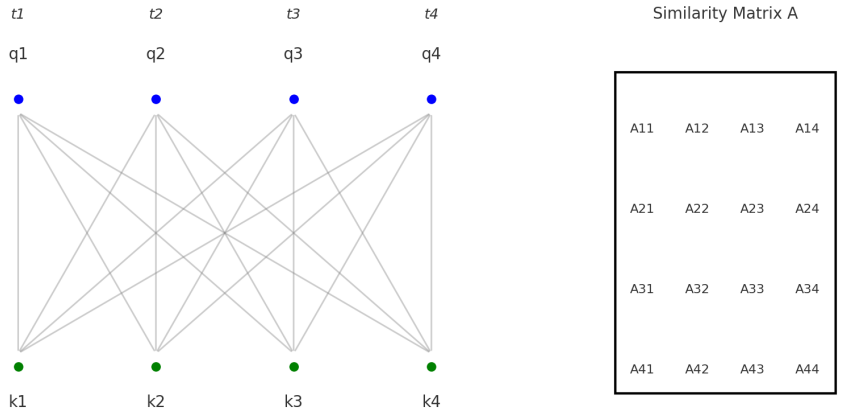

Figure 2: **Pairwise Projection of Query and Key Vectors and Construction of Similarity Matrix A.** Each token t_i is projected into a query vector q_i and a key vector k_i. The Transformer mechanism performs pairwise dot products between all q_i and k_j, filling the similarity matrix A_{ij}. This process is structurally identical to comparing delay-embedded states in phase space. Rather than cognitive attention, the mechanism reconstructs a latent attractor geometry from temporal token relationships, mapping sequences into high-dimensional manifolds.The similarity matrix A_{ij} mirrors phase-space vector alignment (e.g., Takens' delay coordinates), not cognitive selection.

License