INTRODUCING THE TAKENS-BASED TRANSFORMER

*

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

ABSTRACT

This work presents a practical implementation of a Takens-based Transformer. The concept of a transformer based on non-linear dynamical principles was first discussed in *Pairwise Phase Space Embedding in Transformer Architectures*; in that earlier work, the relationship between the previously described "attention" mechanism and pairwise embeddings was shown to be equivalent to Takens' method of delays. The work presented here takes this theory and develops a fully functional, practical implementation of a language-based transformer that explicitly uses Takens' method of delays.

The dominant paradigm in language modelling treats meaning as a property of static semantic vectors, with context retrieved through quadratic-complexity attention mechanisms. A dynamical systems theory approach—where language is treated as a trajectory through semantic phase space rather than a distribution of static points—offers a fundamentally different perspective. The demonstrated Takens-based Transformer (TBT) employs Takens' Delay Embedding Theorem and fully replaces attention with exponential delay-coordinate reconstruction, achieving linear complexity and fixed memory usage. A proof-of-concept was implemented on commodity CPU hardware. Following the development of a complete TBT architecture, the system was quantitatively applied to the Brown Corpus and also demonstrated in both question-answering and generative model configurations.

The results showed that with only 15 million parameters, the system achieves stable convergence (training loss: 1.88, validation: 4.21) on the Brown Corpus. The question-answering and generative text models performed in line with the results of the Brown Corpus study.

These results demonstrate that a practical implementation of Takens' theory in a suitable architecture produces viable language models. Furthermore, the architecture enables topological separation of input, output, and potential intermediate reasoning states using manifold-based semantic separation. This work shows that a fundamentally different architectural lens can be successfully applied to language models. The developed architecture, based on semantic phase space reconstruction rather than attention, can produce functional language models with dramatically different computational properties. The work is offered as a contribution to the theoretical understanding of how language models work, and as an accessible research vehicle for exploring alternatives to the traditional attention paradigm.

Keywords: Language Models, Dynamical Systems, Takens Embedding, Attention Alternatives, CPU Training

* *Citation*: Kevin R. Haylett, "Introducing the Takens-Based Transformer," (December 2025, v1.1), available at https://finitemechanics.com/papers/takens-transformer.pdf

1 Introduction

1.1 The Computational Bottleneck

The transformer architecture has achieved remarkable success in natural language processing, but it carries fundamental computational constraints. The attention mechanism, which compares every token to every other token, scales as $\mathcal{O}(N^2)$ with sequence length. For a 2048-token context, this requires over 4 million comparisons. During generation, cached key-value pairs accumulate linearly, creating ever-growing memory requirements. These constraints have made language modelling research increasingly inaccessible, requiring GPU clusters and institutional compute budgets.

More subtly, the attention mechanism's success raises a theoretical question: *what problem is it actually solving?* The standard explanation is that models need to "look back" at relevant context. However, this describes the goal but not the mechanism. Query-key similarity is one way to implement context retrieval, but is it the only way? Is it even the right way?

1.2 A Dynamical Systems Perspective

The TBT is based on viewing language through a different lens: non-linear dynamics and time series analysis. In dynamical systems theory, Takens' Delay Embedding Theorem provides a method for reconstructing the phase space of a system from a single observable measurement by creating delay coordinates: $[\mathbf{x}(t), \mathbf{x}(t-\tau), \mathbf{x}(t-2\tau), \ldots]$. This technique reveals hidden structure in what appears to be a one-dimensional signal.

From a non-linear dynamical system perspective, language is not a collection of semantic vectors but a trajectory through semantic phase space. Each token is not a location but a control parameter that steers the trajectory. Rather than the sentence "The quick brown fox" being four points in space being averaged, it is a *path* with momentum and direction. Context is not retrieved by comparing vectors; it is *embedded* in the current position on the trajectory.

From this perspective, attention can be seen as an expensive approximation of something simpler—implicitly performing delay embedding by allowing the model to look backward through token history. In this work, we show how attention can be practically and straightforwardly replaced with explicit phase space reconstruction.

1.3 Potential Contributions

The work presented makes three potential contributions: First, a it makes a theoretical contribution demonstrating language can be successfully modelled as trajectories on manifolds, and demonstrate how exponential delay embeddings naturally capture the multi-scale temporal structure of language (phonemes, syntax, narrative). Second, it demonstrates that a TBT can be designed to eliminate attention entirely in favour of explicit delay coordinate construction. As a results the architecture achieves $\mathcal{O}(N)$ complexity and $\mathcal{O}(1)$ memory with no growing cache. Third, as an empirical demonstration it highlights that the application of this approach produces functional language models through MARINA, a 15M parameter proof-of-concept trained on CPU hardware. Following successful implementation training curves, convergence behaviour, and preliminary experiments with channel-separated reasoning are clearly presented.

1.4 What This Work Is Not

Importantly, this work is not claiming to beat or compete with state-of-the-art benchmarks, or replace existing production systems. The goal is significantly narrower and more foundational i.e. to demonstrate that an alternative architecture grounded in dynamical systems, rather than statistical pattern matching, can successfully model language. Furthermore, it is hoped that this will provide a a theoretical contribution that opens new research directions rather than an engineering achievement that advances benchmarks.

2 From Theoretical Conjecture to Practical Implementation

The dynamical systems interpretation of language modelling presented here builds directly on the geometric framework introduced in Finite Tractus: The Hidden Geometry of Language and Thought [Haylett, 2025], where language is formalized as finite trajectories in semantic phase space, with written text as a compressed transduction of underlying acoustic attractors. This earlier work established the equivalence between token sequences and delay-embedded observables, providing the theoretical foundation for explicitly operationalizing Takens' Theorem in place of implicit attention-based reconstruction

In the companion paper Pairwise Phase Space Embedding in Transformer Architectures, the a theoretical reinterpretation of the transformer's attention mechanism was proposed. This earlier work argued that attention could be understood not as a learned relevance-scoring function, but as an implicit form of delay-coordinate reconstruction. In that framing, each query–key interaction acts as a micro-embedding of local temporal history, such that the attention matrix approximates the structure of a Takens-style embedding of the sequence. The initial paper presented this as a conjectural lens, offering conceptual motivation and mathematical parallels. However, despite speculation, it did not *demonstrate* that Takens' method could be operationalised as a full language modelling architecture.

2.1 From Theory to Practice

The work provided here takes that missing step. Rather than treating Takens' theorem as an analogy, the methods used instantiate it directly. The architecture presented replaces attention entirely with explicit exponential delay coordinates and manifolds learned through adaptive projection layers. In doing so, the loop between the earlier theoretical perspective and an implemented system capable of learning linguistic dynamics is closed.

2.2 Initial Significance

The transition from theory to practice is significant for two reasons. First, it establishes that Takens-style embeddings are not merely compatible with language modelling but method of delays is sufficient to support a fully functional next-token predictor using only a single observable (the evolving token stream) and its delayed samples. Second, by demonstrating stable convergence and coherent text generation on the Brown Corpus, the implementation serves as a proof-of-concept validation that the earlier theoretical proposal describes not only an interpretative model of attention, but a practical design principle for alternative sequence architectures.

In this sense, the TBT does not challenge the original theory of attention but completes it. The hypothesis that attention performs a form of implicit phase-space reconstruction is strengthened by the fact that explicit reconstruction can replace attention altogether while retaining meaningful linguistic behaviour. From this perspective the developed TBT can be viewed as a continuation and realisation of the conceptual trajectory initiated by the prior art. The TBT transforms a theoretical insight into an operational system with measurable properties and empirical grounding.

3 Background and Related Work

3.1 Takens' Delay Embedding Theorem

In 1981, Floris Takens proved a fundamental result about reconstructing dynamical systems from time series data [4]. Consider a dynamical system evolving on some manifold $\mathcal{M}$. Even if we can only observe a single measurement $h(\mathbf{x}(t))$—say, temperature readings from a weather station—we can reconstruct the topology of $\mathcal{M}$ by creating delay coordinates:

$$\mathbf{y}(t) = [h(\mathbf{x}(t)), h(\mathbf{x}(t-\tau)), h(\mathbf{x}(t-2\tau)), \ldots, h(\mathbf{x}(t-d\tau))] \tag{1}$$

Under generic conditions, if the embedding dimension d is large enough ($d > 2 \cdot \dim(\mathcal{M})$), the delay embedding $\mathbf{y}(t)$ is diffeomorphic to $\mathcal{M}$. In practical terms: watching how a single measurement changes over time reveals the structure of the hidden system driving those changes.

Importantly, this theorem has been extensively applied in chaos theory, time series prediction, and non-linear signal processing. However, it has not been systematically explored in natural language processing beyond metaphorical references.

3.2 The Attention Mechanism

The transformer's attention mechanism computes context-weighted representations through [5]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

where queries, keys, and values are learned linear projections of input embeddings. This operation compares every query position to every key position, creating an $N \times N$ attention matrix that must be computed, stored, and multiplied by the value matrix. The mechanism's success has been attributed to its ability to model long-range dependencies and to learn which previous tokens are relevant for predicting the next token. However, the quadratic cost has motivated extensive research into more efficient alternatives.

3.3 Linear Attention and Alternatives

Following the success of the attention mechanism, several approaches have attempted to reduce its complexity and computational requirements. These include:

Linear Attention: Reformulating attention as $(Q \cdot V^T) \cdot K$ to achieve $\mathcal{O}(N)$ complexity, but often with degraded performance [9].

State Space Models (S4, Mamba): Treating sequences as linear dynamical systems with learnable state transitions. These achieve linear complexity and have shown strong empirical results [6, 7].

RWKV: Reformulating attention as a recurrent architecture with time-mixing and channel-mixing operations [8].

Fixed-Pattern Attention: Using predetermined attention patterns (e.g., Longformer, BigBird) to sparse the attention matrix [10].

A New Alternative

The non-linear dynamical systems approach presented here differs significantly from these earlier methods in its theoretical foundation. State space models treat the system as a linear dynamical system with state transitions. In this work, language is treated as a non-linear dynamical system where the trajectory of the language signal itself carries information. Prior linear approaches reorganize the attention computation but retain the conceptual framework of query-key matching. In a TBT, the attention framework is removed entirely in favour of delay coordinate reconstruction.

To the author's knowledge, no prior work has systematically applied Takens' theorem to language modelling with exponential delay spacing designed to capture language's multi-scale temporal structure. The closest work, state space models, focuses on learned state transitions rather than explicit manifold reconstruction. This work is positioned as exploring a complementary theoretical lens.

Concurrent work has explored delay embedding interpretations of neural sequence models [3], viewing transformers and state-space models as implicitly reconstructing latent dynamics from observed sequences to infer unobserved meaning in language. This aligns with the dynamical systems perspective here, though that work focuses on inductive biases in time-series prediction tasks. Importantly, the present architecture builds on earlier theoretical links between attention and pairwise phase space embeddings [1, 2], identified via a comparison of Takens' method of delays and the attention mechanism introduced by Vaswani et al. [5].

4 Theoretical Framework

4.1 Language as Manifold Trajectory

From a non-linear dynamical systems perspective the "meaning" of a sequence of tokens is not a property of individual semantic vectors or their averaged combination, but rather the trajectory those tokens trace through a latent semantic manifold. Consider the sentence "The quick brown fox jumps over the lazy dog." In the standard embedding space view, each word maps to a point in $\mathbb{R}^d$, and the sentence meaning emerges from attention-weighted combinations of these points. In our view, the sentence is a *path*: starting at "The," the trajectory is steered toward "quick," curves through "brown," arrives at "fox," and so on. The next-token prediction is determined by the tangent to this trajectory—the direction it's currently heading.

4.2 The Shift from Static Points to Dynamic Trajectories

This shift from static semantic points to dynamic trajectories carries several important implications for how language should be modelled.

4.3 The Importance of Geometry

First, context becomes a geometric property rather than a retrieval problem. In traditional attention-based systems, the model is continually required to search backward through the token history to determine what earlier information remains relevant. In a dynamical systems framing, however, relevance is not something the model must explicitly look up; it is expressed in the current position and momentum of the trajectory. Each new token alters the direction of motion through the manifold, and the entire historical influence is implicitly encoded in the evolving state. Context is therefore a geometric consequence rather than a learned search strategy.

4.4 The importance of the Trajectory

Second, tokens cease to be static carriers of meaning and instead operate as control parameters that steer the trajectory. A token no longer "has" a fixed meaning in isolation; its functional role is to modify the system's evolution. This view captures the well-known linguistic fact that meaning is highly contextual: the effect of a token depends not only on its own embedding but on the state of the system at the moment it is encountered. The manifold trajectory framework provides a precise mathematical expression of that dependency.

4.5 The Landscape of Language

Third, memory emerges as a structural feature of the landscape of language i.e. languages dynamical system manifold, rather than an explicit buffer of past content. In an attention model, memory is stored indirectly via key–value caches or retrieved token relations. In a dynamical system, memory is encoded in the shape of the manifold and the attractors that govern the flow. Trajectories are drawn into certain basins, repelled from others, and constrained by local curvature. This provides a natural explanation for how a model "remembers" patterns without explicitly storing past tokens: structural memory arises from the topology of the learned manifold itself.

Taken together, these implications suggest that linguistic behaviour can be captured by modelling how trajectories unfold in a learned semantic space. The present work operationalises this perspective by constructing such trajectories directly from delay coordinates, rather than approximating them indirectly through attention weights.

4.6 Exponential Delay Embeddings for Multi-Scale Structure

4.7 Language Timescales

Language exhibits structure across several distinct temporal scales, each contributing a different layer of meaning and coherence to a sequence. At the shortest scale, spanning only a few neighbouring tokens, we find the local dynamics that govern syntax, grammatical agreement, and immediate dependencies. These rapid fluctuations determine whether a phrase is well-formed: they capture how determiners attach to nouns, how verbs govern their arguments, and how small adjustments in word order can alter grammaticality. In this near-field regime, linguistic behaviour is governed by fine-grained constraints that evolve token by token.

Beyond this immediate horizon lies an intermediate regime, extending roughly across 10–30 tokens, in which sentences and clauses develop their internal structure. At this scale, the model must integrate information across multiple syntactic units, maintain subject–predicate consistency, handle clause embedding, and track referential cohesion. This medium-range structure reflects the internal organisation of sentences and short passages—elements too dispersed to be handled by purely local mechanisms, yet too compact to be considered long-range discourse.

Finally, language also exhibits dynamics on longer temporal scales, extending across hundreds of tokens. At this scale, the model must track global thematic development, narrative progression, shifts in argument structure, and maintenance or transition of topics. These long-range dependencies provide the overarching coherence that connects paragraphs or entire documents. They evolve slowly compared to token-level syntax, yet remain essential for meaningful text generation.

The coexistence of these temporal layers implies that any architecture aiming to model language effectively must account for patterns that unfold at vastly different rates. The Takens-based approach introduced here does so by employing exponential delay embeddings that naturally encode these multi-scale structures within a single unified mechanism.

A fixed-window delay embedding $[\mathbf{e}_t, \mathbf{e}_{t-1}, \mathbf{e}_{t-2}, \ldots, \mathbf{e}_{t-k}]$ captures recent history but treats all delays equally. We propose exponentially spaced delays:

$$\mathbf{x}_t = [\mathbf{e}_t, \mathbf{e}_{t-1}, \mathbf{e}_{t-2}, \mathbf{e}_{t-4}, \mathbf{e}_{t-8}, \mathbf{e}_{t-16}, \mathbf{e}_{t-32}, \ldots]$$

This logarithmic spacing allocates representation capacity efficiently: dense sampling of recent tokens (syntax-level), sparser sampling of medium history (sentence-level), and very sparse sampling of distant history (discourse-level). For a vocabulary of size V and embedding dimension d, the delay vector has dimension $d \cdot k$ where $k = \lceil \log_2(N) \rceil$ for sequences up to length N.

Crucially, this structure is *fixed*—the delays don't grow with sequence length. The model maintains a circular buffer of recent embeddings and samples from it at exponential offsets. Memory usage is $\mathcal{O}(1)$ regardless of how many tokens have been generated.

4.8 Channel Theory: Topological Control

Standard language models suffer from what we call "manifold collapse": the inability to structurally distinguish between user input, model generation, and internal reasoning. Techniques like chain-of-thought prompting attempt to create separation by inserting explicit text markers ("Let's think step by step..."), but this is a semantic solution to a structural problem.

To operationalise controlled separation between different functional roles within a sequence—such as user inputs, model-generated outputs, and intermediate reasoning—we introduce Channel Theory, a topological mechanism that assigns each token to a distinct region of the semantic manifold. Rather than relying on prompting conventions or special tokens to demarcate roles, Channel Theory embeds these distinctions directly into the geometry of the model's state space.

Each token embedding is augmented with a learned channel vector that marks the semantic "stratum" to which it belongs. For example, tokens originating from user prompts occupy the User Channel, a dedicated subspace structured so that trajectories entering from external input flow along its manifold. Tokens produced by the model during generation inhabit the System Channel, a separate region whose curvature governs how the model formulates responses. Between these two lies the Bridge Channel, a manifold reserved for internal reasoning, intermediate steps, or latent computations that the model performs but does not necessarily expose in its final output.

Because these channels are implemented as orthogonal support vectors, they function as topologically distinct regions rather than merely symbolic labels. A trajectory initiated in the User Channel is constrained to evolve within that region unless it passes through the Bridge Channel, which acts as a formal conduit for transforming user-provided context into system-level output. This ensures that transitions between roles are not arbitrary or accidental: they require a structural traversal of the manifold, rather than a semantic reinterpretation of the same space.

This geometric separation provides a robustness that textual markers or reinforcement-learning strategies cannot guarantee. In prompt-based or RLHF-guided models, role distinctions are semantic conventions. Nothing prevents the system from blending reasoning into output or misclassifying user content as system-generated. In contrast, the topological structure of Channel Theory ensures that the model treats user inputs, internal deliberations, and outward responses as different types of trajectories, each evolving within its own region of the manifold. The result is a principled method for controlling internal reasoning and generation modes that arises from the geometry of the model itself rather than ad-hoc heuristics.

We introduce *Channel Theory* as a topological solution. By augmenting each embedding with orthogonal support vectors, we create parallel manifolds:

$$\mathbf{e}_{\text{augmented}} = [\mathbf{e}_{\text{semantic}}; \mathbf{s}_{\text{user}}; \mathbf{s}_{\text{system}}; \mathbf{s}_{\text{bridge}}] \tag{3}$$

where $\mathbf{s}_{\text{user}}, \mathbf{s}_{\text{system}}, \mathbf{s}_{\text{bridge}} \in \mathbb{R}^c$ are learnable channel vectors. These vectors act as "flags" that keep trajectories separated in phase space:

- *User Channel:* $\mathbf{s}_{\text{user}} = [1, 0, 0]$, input tokens live here
- *System Channel:* $\mathbf{s}_{\text{system}} = [0, 1, 0]$, generated output lives here
- *Bridge Channel:* $\mathbf{s}_{\text{bridge}} = [0, 0, 1]$, intermediate reasoning lives here

The manifold projection learns to treat these channels as topologically distinct regions. A trajectory in the User Channel cannot drift into the System Channel unless explicitly bridged through the transformation. This provides a *mathematical guarantee* of role separation that reinforcement learning cannot offer—the channels are structurally different regions of phase space.

4.9 The Manifold Projection Layer

The raw delay vector is high-dimensional and sparse (most historical positions are zero-padded). We need a learned projection that maps this structured input onto the dense semantic manifold:

$$\mathbf{z}_t = \text{LayerNorm}(W_P \cdot \mathbf{x}_t) \tag{4}$$

where $W_P \in \mathbb{R}^{d_{\text{model}} \times d_{\text{delay}}}$ is the Adaptive Takens Layer. This matrix learns which timescales are relevant for prediction. After training, we can inspect W_P to understand what temporal patterns the model considers important—a form of interpretability unavailable in attention-based models.

5 Theoretical Considerations

Revealing the Topological Landscape of Language

The dynamical systems perspective adopted here yields a number of testable predictions about the geometric structures a Takens-based language model should learn. These predictions are not metaphorical descriptions of linguistic behaviour; they are claims about the actual topology of the learned phase space — the curvature, basins, attractors, and separatrices that arise mathematically when the model reconstructs trajectories from delay coordinates. In this framing, the "semantic manifold" becomes a literal object: a high-dimensional surface on which linguistic sequences carve out paths governed by the local geometry encoded in the model's weights.

5.1 Distinct Trajectories: Tasks requiring precise recall form narrow, tubular attractors.

When a model must reproduce specific sequences — such as fixed phrasing, quotations, or repeated patterns — the corresponding trajectories should collapse onto thin isoclines of the manifold. These behave like "memory fibres": sharply defined curves along which the dynamics must evolve with minimal deviation. In this region of the manifold, local curvature is high and stability is tight; small perturbations should return the trajectory to the tube, much like a stable orbit. In contrast, tasks involving generalisation or paraphrasing should produce broader basins of attraction, where many nearby trajectories converge to similar semantic outcomes even if their precise paths differ. This provides a geometric criterion for distinguishing rote memorisation from meaningful generalisation: the former occupies thin, rigid attractors, whereas the latter spans wide, gently-sloped basins.

5.2 System Attractors: Language Forms Attractors.

Non-linear dynamical systems are characterised by system attractors and sensitivity to initial conditions. System attractors are the regions of semantic phase space that the trajectories follow. From this perspective the semantic phase space should be made up of district regions where trajectories will flow. In addition non-linear dynamic systems are characterised by being sensitive to initial conditions. It is posited that this may be evidenced in observations of the models behaviour.

5.3 Channel separation should produce topologically distinct regions of the manifold.

If user input, internal reasoning, and model output are represented by orthogonal channel vectors, then the learned manifold should contain separate but adjacent regions corresponding to these functions. This implies real geometric separation: trajectories originating in the User Channel should lie on a sheet or branch of the manifold that cannot easily drift into the System or Bridge regions. The only way for a trajectory to transition between these roles is to follow paths encoded by learned transformations through the Bridge Channel. This creates a dynamically enforced separation: the phase-space geometry itself preserves the boundaries. The model cannot "accidentally" reveal its internal reasoning in its output, because doing so would require crossing a topological barrier the system has not learned to traverse.

Together, these predictions frame language modelling as the study of structured flows on a learned manifold. They suggest that the TBT does not merely provide an efficient approximation of attention, but creates a mathematically grounded, dynamically interpretable landscape in which linguistic phenomena occupy identifiable topological structures. Subsequent sections provide preliminary empirical observations consistent with these predictions.

6 Static Embeddings and Dynamical Embeddings: A Geometric Reframing

6.1 The Limits of Static Semantic Vectors

Traditional transformer architectures treat word embeddings as *static points* in a fixed semantic space. Each token w is mapped to a vector $e_w \in \mathbb{R}^d$, a location determined by large-scale co-occurrence statistics. The classical illustration of this approach is the well-known analogy:

$$e_{\text{king}} - e_{\text{man}} + e_{\text{woman}} \approx e_{\text{queen}}.$$

This relation is often interpreted as evidence of deep linguistic structure encoded in geometry. Yet it arises entirely from static correlations in the embedding space.

In this view, the embedding space is inert: meaning is treated as a property of *locations*. Attention mechanisms must continually "look backward" to reconstruct context, because context is not inherent in the representation. The architecture compensates for the fact that static embeddings cannot move.

This perspective, though powerful, obscures the fundamentally *temporal* character of language.

6.2 Words as Motions: The Dynamical Perspective

A TBT reframes the problem by treating words not as points but as *trajectories*. The model observes a token stream as a temporal signal and reconstructs its state through delay coordinates. For a sequence (w_t), the model constructs

$$x_t = \big[e(w_t),\ e(w_{t-\tau}),\ e(w_{t-2\tau}),\ e(w_{t-4\tau}),\ \dots\big],$$

a vector representing not an isolated token but a *window of motion* through semantic space. Takens' Theorem guarantees that such windows can reconstruct the underlying dynamics of the system.

Under this interpretation:

- the transition man $\to$ king is not a vector subtraction but a dynamical evolution observed across many contexts, and
- the transition woman $\to$ queen is a parallel flow in the reconstructed manifold.

The canonical analogy is thus not a static algebraic identity but the projection of *two homologous trajectories*.

6.3 How Classical Analogies Emerge Dynamically

Consider the sentences:

> "The king ruled the land ..."
> "The queen addressed the people ..."

In a static embedding model, similarities are derived from pointwise relations. In MARINA, the model instead observes how the embedding trajectory evolves in each context:

$$x_{\text{man}} \mapsto x_{\text{king}}, \qquad x_{\text{woman}} \mapsto x_{\text{queen}}.$$

Each x_w is a delay-embedded slice of the surrounding linguistic environment. The structure the model learns is therefore not the static relation

$$e_{\text{king}} - e_{\text{man}} \approx e_{\text{queen}} - e_{\text{woman}},$$

but the deeper dynamical invariants:

$$T(\text{man}) \to T(\text{king}), \qquad T(\text{woman}) \to T(\text{queen}),$$

where T is the learned transition operator on the reconstructed semantic manifold.

The classical vector analogy is thus a *shadow* of two structurally equivalent flows.

6.4 Why the Distinction Matters

Static embeddings describe language as a constellation of points, whereas dynamical embeddings describe language as a field of flows. In the static view, analogies appear as algebraic coincidences, meaning is inferred from pointwise relations, and context must be reconstructed through attention mechanisms. In the dynamical view, however, analogies arise from structural invariants of trajectories, meaning is encoded by curvature, direction, and flow, and context is inherent in the reconstructed state. One may think of the contrast this way: a static embedding is a map of cities; a dynamical embedding is the network of roads, currents, and flows that connect them. The relation between man–king and woman–queen stands out not because the points resemble each other, but because the motions connecting them share geometric structure.

6.5 Bridging to the Architecture

This reframing prepares the ground for the TBT. If meaning lies not in discrete token locations but in the *trajectories they induce*, then delay-coordinate reconstruction becomes not an alternative to attention but a principled foundation for modelling language as a dynamical system. The architecture introduced in the following sections is designed to capture precisely these flows.

7 The TBT Architecture (MARINA)

7.1 Overview

The specific TBT Architecture used is named as sManifold-Aware Reconstruction and Inference Network Architecture (MARINA). MARINA's architecture implements the TBT by translating the theoretical framework of exponential delay embeddings into a coherent, operational language model. Rather than relying on attention mechanisms or key–value caches, MARINA reconstructs the evolving semantic state directly from structured samples of its recent history, projecting these reconstructed states onto a learned manifold from which predictions are generated. The architecture is intentionally compact and conceptually modular, allowing each component to reflect one aspect of the underlying dynamical-systems interpretation. MARINA consists of four tightly integrated subsystems.

1. *MVec Encoder:* Converts tokens to embeddings with channel vectors
2. *Exponential Takens Embedding:* Constructs delay coordinates
3. *Adaptive Manifold Projection:* Maps delays to latent manifold
4. *Channel-Aware Output:* Separates prediction heads by channel

The resulting delay vector is then processed by the adaptive manifold projection layer, which compresses the high-dimensional delay coordinates into a dense semantic representation. This layer learns the geometry of the semantic manifold itself: its curvature, attractors, and transitional regions. Through this projection, the system transforms raw historical embeddings into a smooth, navigable state that represents the trajectory's present position and allows for coherent evolution through the manifold.

Together, these components form a unified architecture in which linguistic behaviour emerges from the geometric evolution of trajectories in a reconstructed semantic phase space. MARINA operationalises the central claim of this work: that explicit delay-coordinate reconstruction can successfully replace attention, yielding a model that is both interpretable in its dynamics and efficient in its computational requirements. Figure 1 compares the architecture to a standard transformer.

7.2 Multi-Vector (MVec) Encoding

The first is the MVec encoder, which enriches each token embedding with positional information and channel vectors that specify whether a token belongs to the user input, the system's output, or internal reasoning. This encoding establishes the initial coordinates of each token in the semantic manifold and embeds the topological separations introduced earlier in the theoretical framework.

Each token t is encoded as:

$$\mathbf{e}_t^{\text{full}} = [\text{Embed}(t); \text{Channel}(t); \text{Position}(t)] \tag{5}$$

where:

- $\text{Embed}(t) \in \mathbb{R}^{d_{\text{model}}}$: semantic embedding
- $\text{Channel}(t) \in \mathbb{R}^{c}$: one-hot channel indicator
- $\text{Position}(t) \in \mathbb{R}^{p}$: sinusoidal position encoding (optional)

The channel vector explicitly marks which manifold the token belongs to. During training, we set:

- User input: Channel = $[1, 0, 0, \ldots]$
- System output: Channel = $[0, 1, 0, \ldots]$
- Bridge/reasoning: Channel = $[0, 0, 1, \ldots]$

7.3 Exponential Delay Embedding

The encoded embeddings feed into the exponential Takens embedding mechanism, which forms the core of the architecture. Here, MARINA collects a sequence of delayed samples from the token history, spaced at exponential intervals. These delays serve as structured observables from which the model reconstructs its current phase-space location. By using exponential spacing, the system simultaneously captures fine-grained syntactic details, intermediate sentence-level dependencies, and long-range narrative influences, all within a fixed-size memory footprint. At each time-step t, we construct:

$$\mathbf{x}_t = [\mathbf{e}_t, \mathbf{e}_{t-1}, \mathbf{e}_{t-2}, \mathbf{e}_{t-4}, \mathbf{e}_{t-8}, \ldots, \mathbf{e}_{t-2^k}] \tag{6}$$

Implementation uses a circular buffer B of size 2^{k+1} where $k = \max_{\text{delay\_power}}$. For a 256-token maximum context with delays $[1, 2, 4, 8, 16, 32]$, we need a 64-element buffer. When predicting token t:

FOR $i \in \{0, 1, 2, 4, 8, 16, 32\}$ STATE $\mathbf{x}_t[i] = B[(t-i) \mod |B|]$ ENDFOR

The buffer is updated at each step: $B[t \mod |B|] \leftarrow \mathbf{e}_t$

This structure ensures:

- *Fixed memory:* $\mathcal{O}(2^k \cdot d)$ regardless of sequence length
- *Fast access:* $\mathcal{O}(k)$ lookups per timestep
- *No quadratic growth:* Unlike KV-caching in transformers

7.4 Explicit Phase-Space Reconstruction via Delay Embedding

We replace attention-based context aggregation with an explicit reconstruction of semantic phase space using delay-coordinate embedding. Rather than computing pairwise token interactions, the model reconstructs the local state of the system directly from its recent trajectory.

Let $\mathbf{x}_{b,t} \in \mathbb{R}^d$ denote the embedding of token t in batch element b. For a fixed set of delay offsets $\{\tau_1, \ldots, \tau_M\}$, we construct the raw delay-coordinate vector

$$\tilde{\mathbf{z}}_{b,t} = \left[\mathbf{x}_{b,t}, \mathbf{x}_{b,t-\tau_1}, \mathbf{x}_{b,t-\tau_2}, \ldots, \mathbf{x}_{b,t-\tau_M}\right] \in \mathbb{R}^{(M+1)d}, \tag{7}$$

with zero padding applied when $t < \tau_i$.

This vector is the explicit Takens embedding of the token stream: it represents a local reconstruction of the system's phase-space neighborhood using only past observations. Temporal structure is therefore made explicit rather than inferred implicitly through attention weights.

The delay offsets are chosen to be exponentially spaced, enabling the model to sample multiple temporal scales—local syntactic structure, sentence-level organization, and longer-range discourse—while maintaining a fixed and linear memory footprint.

—

7.5 Adaptive Manifold Projection and State Representation

The delay-coordinate vector $\tilde{\mathbf{z}}_{b,t}$ is high-dimensional and sparse. To obtain a compact and usable state representation, it is mapped onto a learned semantic manifold by a single affine projection followed by normalization:

$$\mathbf{z}_{b,t} = \mathrm{LayerNorm}(\mathbf{W}_p \tilde{\mathbf{z}}_{b,t} + \mathbf{b}_p), \tag{8}$$

where

$$\mathbf{W}_p \in \mathbb{R}^{d_{\text{out}} \times (M+1)d}, \qquad \mathbf{b}_p \in \mathbb{R}^{d_{\text{out}}}. \tag{9}$$

The projection matrix $\mathbf{W}_p$ learns how information from different temporal delays should be combined, effectively compressing the reconstructed phase-space coordinates into a lower-dimensional manifold. No pointwise nonlinearity is required at this stage: the reconstruction and compression are geometric rather than heuristic.

The bias term $\mathbf{b}_p$ serves only to provide *reference positioning* of the learned manifold within projection space. It introduces no temporal structure and plays no role in the delay-coordinate reconstruction itself.

The normalized vector $\mathbf{z}_{b,t}$ represents the model's current position on the semantic manifold and is the quantity passed to downstream temporal mixing and decoding components. Because context is reconstructed explicitly through delay embedding, no key–value memory, attention weights, or pairwise token comparisons are required.

7.6 Channel Aware Decoding

Finally, MARINA employs a channel-aware output module that generates predictions while respecting the topological separation encoded by the channel vectors. Each channel—user, system, and bridge—has its own output head, ensuring that trajectories originating in one region of the manifold cannot drift into another without passing through an explicitly learned transformation. This architectural choice enforces structural identity separation, allowing the model to reason internally without unintentionally revealing intermediate steps or blending user and system behaviours.

Instead of a single output projection, we maintain separate prediction heads for each channel:

$$\text{logits}_{\text{user}} = W_{\text{out\_user}} \cdot \mathbf{z}_t \tag{10}$$

$$\text{logits}_{\text{system}} = W_{\text{out\_system}} \cdot \mathbf{z}_t \tag{11}$$

$$\text{logits}_{\text{bridge}} = W_{\text{out\_bridge}} \cdot \mathbf{z}_t \tag{12}$$

During training, we compute loss only on the active channel for each token. During generation, we sample from the appropriate channel's distribution based on the current generation mode (user, system, or bridge).

This architectural separation prevents the model from confusing "what the user said" with "what I'm generating"—a structural guarantee that prompting alone cannot provide.

8 Model Parameterisation and Configurable Architecture

While the previous sections describe the general structure of the TBT, the MARINA system is not a single fixed model but a configurable architectural template. Each component can be adjusted to create models of different sizes, capacities, and computational footprints. This modularity allows the system to serve both as a theoretical platform for studying phase-space representations of language and as a practical toolkit for building resource-conscious models.

8.1 Embedding and Channel Dimensions

The MVec encoder allows flexible selection of the semantic embedding dimension model, the size of the channel vectors, and the positional encoding scheme. Larger embedding dimensions increase the expressive capacity of the manifold projection, whereas smaller dimensions enable lightweight, CPU-friendly models. The number and size of channel vectors determine the degree of topological separation between user, system, and reasoning spaces.

Delay Schedule Configuration: A central parameter is the structure of the exponential delay schedule. The model designer may choose how many delays to include and where to place them. Shorter schedules emphasise local syntax and fast dynamics, while deeper schedules capture broader narrative structure. Because the Takens embedding is fixed rather than learned, this choice directly shapes the model's effective receptive field without altering runtime memory usage.

Manifold Projection Depth and Width:

The adaptive projection layer can be configured with different hidden dimensions and nonlinearities. Deeper projections allow the model to learn more complex manifold geometries, while shallow projections offer interpretability and computational simplicity. These parameters control how the high-dimensional delay vector is compressed onto the learned semantic manifold.

Temporal Mixing Layers:

MARINA can incorporate an arbitrary number of temporal mixing blocks. Each block contributes additional nonlinearity and expressive power, enabling the architecture to capture multi-scale interactions more deeply. Fewer blocks yield faster training and easier interpretability; more blocks extend capacity without increasing memory complexity.

Vocabulary and Tokenisation Choices:

The system supports both word-level and subword vocabularies. The choice affects the sequence length, memory demands, and linguistic granularity of the reconstruction. For lightweight models, a curated vocabulary or domain-specific lexicon may be preferable, while larger-scale models benefit from subword methods.

Training Hyperparameters and Sequence Length: Although training choices are not intrinsic properties of the architecture, they significantly affect performance. Batch size, learning rate schedules, optimisation algorithms, and the maximum sequence length can all be tuned independently. Notably, the model's memory footprint remains constant regardless of sequence length, a property that allows experimentation with extremely long contexts without altering the architecture.

8.2 Channel Configuration and Reasoning Modes

Finally, the number of channels and their associated output heads can be customised. While this work focuses on the three-channel (user, system, bridge) configuration, the architecture supports additional channels for specialised tasks such as summarisation, retrieval, system reasoning layers, or isolated memory streams.

Together, these degrees of freedom mean that MARINA should not be viewed as a single model but as a family of Takens-based architectures, each instance tuned to a particular research question or computational setting. The experimental model presented in Section 5 represents only one such configuration, chosen for accessibility and ease of replication rather than optimal performance.

8.3 Examples and Practical Implementation

The flexibility of these design parameters allows MARINA to be instantiated in a wide variety of forms, each suited to a different investigative purpose. To demonstrate both the generality and practical viability of the TBT, we present a sequence of model configurations rather than a single experimental case. Each instantiation highlights a different aspect of the architecture and reveals distinct properties of the underlying manifold dynamics.

8.4 Application to the Brown corpus

Our first example is a model trained on the Brown Corpus, a balanced sample of mid-century American English. This experiment establishes that the architecture can learn broad linguistic structure, converge stably, and generate coherent text using only delay-coordinate reconstruction, without attention or recurrence in the traditional sense. It demonstrates the baseline linguistic competence achievable by explicit phase-space reconstruction.

8.5 Application to a structured question-answer dataset

The second model applies MARINA to a structured Question–Answer task built from curated Solar System QA pairs. Unlike the Brown Corpus, which emphasises open-form generative behaviour, this dataset isolates factual reasoning and short-range semantic control within a well-defined knowledge domain. Here, channel separation and manifold topology play a more visible role, as the system must maintain boundaries between user queries, internal reasoning pathways, and final answers.

8.6 Application to structured narration

Finally, we present a third example: a generative model trained on an engineered mythopoetic corpus derived from the Corpus Ancora. This dataset is deliberately designed to exhibit strong stylistic coherence, conceptual recursion, symbolic motifs, and long-form narrative structure. The model's success on this material demonstrates that MARINA can learn not only factual or syntactic patterns, but also high-level aesthetic and semantic regularities. In this domain, the manifold geometry becomes especially expressive: it must capture thematic arcs, recursive symbolic structures, and stylistic constraints across extended trajectories. This experiment illustrates MARINA's ability to generate novel text that preserves the meaning, rhythm, and conceptual architecture of a curated mythos.

Taken together, these three instantiations show that MARINA is not a single model but a configurable dynamical framework capable of supporting diverse language behaviours. The Takens-based approach scales gracefully across open-ended generation, structured reasoning, and stylistically constrained narrative synthesis. With this range of configurations established, we now turn to the empirical behaviour of the first model.

Table 1: Configurable Parameters in the MARINA Architecture

Parameter Group	*Description and Options*
Embedding Dimension	Semantic embedding size d_{model}; controls manifold resolution and representational capacity. Typical range: 128–2048.
Channel Configuration	Number of channels and channel-vector dimension; determines topological separation of user, system, and reasoning trajectories. Custom channels may be added.
Delay Schedule	Choice of exponential delays $[1, 2, 4, 8, \ldots, 2^k]$; governs the temporal scales captured by reconstruction. Number of delays and maximum delay power k are configurable.
Manifold Projection Depth	Number of projection layers and hidden dimension d_{hidden}; controls the geometric complexity of the learned manifold.
Temporal Mixing Layers	Count and width of feed-forward mixing blocks; increases model expressiveness without altering memory complexity.
Tokenisation Scheme	Selection of vocabulary construction (word-level, subword/BPE, domain-specific lexicons). Affects sequence length and semantic granularity.
Position Encoding	Choice of positional signals (sinusoidal, learned, or none). Not strictly required but may improve stability.
Sequence Length	Maximum training sequence length; independent of memory footprint due to fixed-size delay buffer. Enables experimentation with long contexts.
Training Hyperparameters	Learning rate schedule, batch size, optimiser (AdamW, SGD, etc.). These influence performance but do not alter architecture.
Output Heads	Channel-dependent output projections; may be extended for specialised generation tasks or multi-manifold reasoning streams.

9 Experimental Examples

The previous sections established MARINA as a configurable dynamical architecture grounded in Takens-style delay reconstruction and manifold projection. We now turn to the practical realisation of this framework through three distinct experimental examples. Each experiment instantiates the same underlying architecture but selects different parameterisations, datasets, and channel configurations to test specific theoretical predictions about the geometric and dynamical behaviour of language.

9.1 Objectives and Implementation

Our goal is not to pursue benchmark superiority, but to demonstrate that the TBT is capable of modelling linguistic behaviour across multiple regimes: general-language distributional structure, domain-specific factual reasoning, and highly stylised mythopoetic generation. Each experiment highlights a different aspect of the manifold-trajectory viewpoint and provides empirical evidence about how delay embeddings interact with temporal structure, semantics, and narrative patterns.

Model 1, trained on the Brown Corpus, evaluates whether explicit phase-space reconstruction can learn the broad statistical regularities of natural language. This experiment serves as a baseline for dynamical stability, convergence behaviour, and general linguistic competence.

Model 2, trained on a curated set of Solar System Question–Answer pairs, tests the architecture's ability to represent structured reasoning pathways within bounded semantic domains. Here, channel separation and manifold topology play a more prominent role, as the model must maintain clear distinctions between user prompts, internal reasoning trajectories, and final responses.

Model 3: is a generative system trained on the Corpus Ancora, an engineered mythopoetic text designed to exhibit recursive motifs, symbolic structure, and stylistic coherence. This experiment examines whether the manifold learned by MARINA can accommodate long-range thematic arcs, conceptual repetition, and structured creative expression, revealing its capacity not just for factual reasoning but for controlled generative synthesis.

Together, these examples demonstrate that the TBT is not a narrow substitute for attention, but a general-purpose framework for modelling language as a dynamical system. Each dataset activates different regions of the semantic manifold and probes different temporal scales, providing complementary views of the architecture's behaviour.

We now describe each experiment in turn, beginning with the Brown Corpus.

10 Model 1: Brown Corpus: General Linguistic Dynamics

The Brown Corpus serves as the first empirical test of the TBT, providing a balanced and heterogeneous sample of English from which to observe the model's capacity to learn broad linguistic structure. Its diversity—spanning news, fiction, essays, and academic writing—makes it an ideal benchmark for evaluating whether a dynamical architecture can capture the general curvature of a natural language manifold. More importantly, the Brown Corpus offers a setting in which no specialised domain knowledge or controlled structure can assist the model; it must learn language as a trajectory rather than a catalogue of facts.

For this experiment, MARINA was configured in a compact form, using a modest embedding dimension, a lightweight manifold projection, and an exponentially spaced delay schedule. Only high-level configuration details are included here; the complete parameter listing and training configuration are provided in Appendix **??**. The goal was not to compete with large-scale attention-based models but to assess whether explicit delay-coordinate reconstruction is sufficient to support stable convergence and coherent generative behaviour under realistic corpus conditions.

Training proceeded smoothly on standard CPU hardware. The model demonstrated stable convergence, with loss curves following a characteristic shape consistent with models that form coherent internal manifolds. Although the absolute loss values reflect the model's small scale and limited capacity, the qualitative behaviour is more significant: the model learned to produce well-formed sentences, maintained local syntactic consistency, and generated text exhibiting recognisable Brown-like stylistic structure. Representative samples are included below, with extended generations provided in Appendix **??**.

These results provide the first concrete evidence that a Takens-based architecture can support general-purpose language modelling. The model did not rely on attention to retrieve or compare tokens; instead, it reconstructed a latent state from delay embeddings and evolved this state across the manifold. The coherence of the generated text suggests that the model successfully learned a geometric representation of linguistic flow, validating the central theoretical claim that language behaves as a dynamical trajectory rather than a pointwise configuration.

The Brown Corpus experiment therefore serves as the foundation upon which the following sections build. It establishes that the architecture can learn generic English, that it converges stably, and that the reconstructed manifold is sufficiently expressive to support coherent linguistic motion. The subsequent experiments extend this result into more specialised domains, testing the model's behaviour under structured question-answer tasks and long-range stylistic generation. Full loss curves, training logs, and parameter details are provided in the appendices.

Dataset and Training Setup	
Dataset	Brown Corpus (55,542 vocabulary, $\sim$1M words)
Chunking strategy	Sliding windows of 256 tokens (stride 128, 50% overlap)
Objective	Next-token prediction with cross-entropy loss
Loss function	$\mathcal{L} = -\sum_t \log P(\text{token}_t \mid \text{delay\_context}_{t-1})$
Optimisation	
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$)
Learning rate	3×10^{-4} with cosine annealing
Batch size	8 sequences
Gradient clipping	1.0
Early stopping	Patience of 10 epochs (based on validation loss)
Hardware and Training Time	
Hardware	Intel i7, 32 GB RAM, no GPU
Training time	$\sim$14–17 minutes per epoch

Table 2: Dataset preparation and training configuration for the Brown Corpus experiment.

Table 3: Architecture comparison

Component	*Standard Transformer*	*MARINA (TBT)*
Context mechanism	Multi-head attention	Exponential delays
Complexity per token	$\mathcal{O}(N^2)$	$\mathcal{O}(\log N)$
Memory growth	$\mathcal{O}(N)$ KV-cache	$\mathcal{O}(1)$ fixed buffer
Context retrieval	Query-key similarity	Phase space embedding
Hardware requirement	GPU clusters	CPU sufficient
Interpretability	Attention weights	Manifold geometry

10.1 Training

Having established the conceptual foundation, we now detail the Brown Corpus training experiment. We trained MARINA on the Brown Corpus, a balanced collection of American English texts from 1961. The corpus represents multiple genres (press, fiction, government documents, academic writing) and serves as a reasonable test of general language modelling capability without the scale or computational requirements of modern web-scale datasets.

Dataset Statistics	
Vocabulary size	55,542 words
Training samples	7,137 sequences (256 tokens each)
Validation samples	792 sequences
Total training tokens	$\sim$1.8M
Total validation tokens	$\sim$200K
Model Configuration	
Total parameters	14,987,480 ($\sim$15M)
Embedding dimension	512
Hidden dimension	1024
Delay coordinates	$[1, 2, 4, 8, 16, 32]$
Channels	3 (user, system, bridge)
Layers	4 temporal mixing blocks

Table 4: Dataset statistics and model configuration for the Brown Corpus experiment.

Training Dynamics	
Training epochs	44 (early stopping)
Initial loss (Epoch 1)	6.74 (training), 6.03 (validation)
Final loss (Epoch 44)	1.88 (training), 4.21 (validation)
Best validation loss	4.21 (Epoch 34)
Total improvement	4.85 reduction in training loss

Table 5: Training dynamics for the Brown Corpus TBT.

The training dynamics, shown in Figure 2, reveal several notable properties of the model's behaviour. The training loss decreases monotonically across the full 44-epoch run, exhibiting no instability or divergence, while the validation loss settles into a stable plateau around 4.2 after approximately 20 epochs. Importantly, the model shows no signs of catastrophic overfitting: although training loss continues to decline, the validation curve remains stable, indicating that the learned manifold generalises well to unseen Brown Corpus sequences. The cosine annealing learning rate schedule produces visible periodic improvements, corresponding to the small oscillations seen in both training and validation loss. Overall, the convergence rate is consistent with a two-phase learning process: the model reduces loss by roughly 0.11 per epoch on average, with rapid early improvements exceeding 0.5 loss per epoch, followed by slower refinement in the later stages (0.02 loss per epoch). This pattern suggests that the Takens-based architecture first captures coarse linguistic structure before refining finer-grained semantic and syntactic details.

10.2 Qualitative Observations

Vocabulary Coverage: The model successfully learns representations for a 55K vocabulary—substantially larger than many small-scale experimental models. It handles function words, content words, rare terms, and proper nouns without special tokenization schemes.

Generalization behaviour: Inspecting generations (not shown for space), the model produces grammatically coherent text that maintains local coherence over 10-20 tokens. It occasionally produces semantically plausible but factually incorrect statements—a behaviour consistent with learning statistical structure without grounding knowledge.

Training Stability: CPU training remained stable throughout 44 epochs with no numerical instabilities, gradient explosions, or optimization failures. The architecture appears robust to the precision limitations of CPU-based training.

10.3 Results in Context

Validation Loss

The final validation loss of 4.21 corresponds to a perplexity of approximately 67. This is reasonable for a 15M-parameter model trained on a word-level vocabulary of 55,000 tokens. For comparison, GPT-2 Small (117M parameters) reports perplexities of 40-50 on similar English corpora, despite being nearly an order of magnitude larger and using more efficient byte-pair encoding.

Parameter Efficiency:

The 15M parameter count prioritizes accessibility over performance. Modern transformers achieve better metrics with 100M-1B parameters but require GPU training and cannot run on edge devices. By contrast, a transformer with 2048-token context would require a KV-cache of 2048 embeddings per layer per head—orders of magnitude larger than our fixed buffer.

10.4 Memory and Compute Characteristics

Memory Usage	
Fixed buffer size	64 embeddings $\times$ 512 dims = 32,768 floats ($\sim$131 KB)
Parameter storage	15M parameters ($\sim$60 MB)
Total inference memory	$<$100 MB including activations
Compute Profile	
Training time	$\sim$14–17 minutes per epoch (Intel i7 CPU)
Training throughput	$\sim$1,200 tokens/second
Inference speed	$\sim$500 tokens/second (CPU, unoptimized)

Table 6: Memory footprint and compute characteristics of the Brown Corpus TBT.

11 Solar System Question and Answer Experiment: Precision Memory Through Tubular Attractors

11.1 Motivation and Dataset Rationale

The preceding experiments demonstrated the TBT's capacity for learning broad semantic structures in natural language (Brown Corpus) and compositional thematic patterns (Corpus Ancora). These tasks required the model to navigate wide attractor basins that support generalization across novel phrasings and contexts. However, language models must also support a fundamentally different mode of operation: precise factual retrieval.

The Question-Answer paradigm represents a distinct geometric challenge. Unlike generative text, where many valid continuations exist, or mythopoetic narrative, where thematic variations are desirable, factualquestion and answerdemands point-to-point mappings: a specific question must reliably produce a specific answer. From the manifold perspective, this suggests a different topological structure—not broad basins that accommodate variation, but narrow channels that enforce precision.

To probe this regime, we constructed a Solar System Question-Answer dataset comprising approximately 500question and answerpairs covering planetary properties, orbital mechanics, physical characteristics, and astronomical facts. Questions ranged from simple factual queries ("What is the largest planet in the solar system?") to more complex multi-part questions ("How does Jupiter's mass compare to Earth's, and what effect does this have on its gravity?"). Answers varied in length from single-word responses to multi-sentence explanations.

The dataset provides several analytical advantages:

1. *Well-defined ground truth*: Each question has a specific correct answer
2. *Bounded semantic domain*: All content relates to solar system astronomy
3. *Vocabulary constraint*: Limited to 2,030 words, creating a compact semantic space
4. *Varying complexity*: Questions span multiple difficulty levels and answer lengths

This experiment tests a specific theoretical prediction from Section 5.1: that tasks requiring precise recall should form narrow, tubular attractors—"memory fibres" along which trajectories must evolve with minimal deviation.

11.2 Experimental Design: Progressive Data Repetition

To investigate the emergence and refinement of geometric memory fibres as a function of repeated exposure, we designed a progressive repetition protocol in which identical question–answer pairs were systematically reintroduced during training. By increasing repetition across successive experiments while holding all other factors constant, this protocol isolates the geometric effects of repetition from conventional optimization dynamics.

Table 7: Progressive repetition experiments for the Solar Systemquestion and answertask

Experiment	*Repeats*	*Training Samples*	*Validation Samples*	*Primary Purpose*
Baseline	1×	463	51	Establish baseline performance and initial attractor discovery
Repetition I	2×	926	102	Observe attractor formation and initial tube refinement
Repetition II	3×	1,388	154	Track progressive narrowing of tubular attractors
Repetition III	4×	2,056	205	Observe asymptotic approach to precision limit

All experiments used identical model architecture (1.23M parameters), vocabulary (2,030 words), and training hyperparameters. The only variable was the number of times eachquestion and answerpair appeared in the training data.

Hypothesis: If the TBT learns geometric structure rather than statistical patterns, repeated exposure to identicalquestion and answerpairs should progressively narrow the trajectory tubes connecting questions to answers. This narrowing should manifest as:

1. Decreasing training loss (tighter trajectory constraint)
2. Improving validation loss (tube formation aids generalization initially)
3. Decreasing train/val gap (tubes become so precise that validation follows training)
4. Asymptotic behaviour (approaching the precision limit of the representation)

11.3 Results: Formation of Memory Fibres

The progressive repetition experiment revealed a striking pattern of geometric refinement, consistent with the formation of increasingly narrow tubular attractors.

Training Loss Progression

The training loss trajectory exhibits two distinct phases:

Phase 1: Rapid Tube Formation (1× → 2×) The transition from single-pass to double-pass training produced an 85% reduction in training loss (1.02 → 0.16). This dramatic improvement indicates that the model transitioned from learning

Repetitions	*Final Training Loss*	*Reduction from Previous*	*Cumulative Reduction*
1×	1.0174	—	—
2×	0.1553	85%	85%
3×	0.1297	16%	87%
4×	0.1127	13%	89%

Table 8: Training loss progression across repetitions showing two-phase learning: rapid tube formation (1× → 2×) followed by progressive narrowing (2× → 4×).

approximate Q→A associations to carving precise trajectories through the manifold. The first exposure discovers that questions and answers can be connected; the second exposure refines those connections into stable pathways.

Phase 2: Progressive Tube Narrowing (2× → 4×) Subsequent repetitions showed diminishing but consistent improvements (16% and 13% reductions). This asymptotic approach suggests the model is approaching the precision limit imposed by the embedding dimension and manifold curvature. Each additional pass further constrains the trajectories, but the returns diminish as the tubes become increasingly narrow.

Validation Loss and Generalization Dynamics

Repetitions	*Final Validation Loss*	*Improvement*	*Train/Val Gap*
1×	2.9688	—	1.95
2×	0.6535	78%	0.50
3×	0.1984	70%	0.07
4×	0.1259	37%	0.01

Table 9: Validation loss and train/val gap across repetitions. The gap collapse from 1.95 to 0.01 indicates the formation of shared geometric structures accessed by both training and validation trajectories.

The validation behaviour reveals the most theoretically significant aspect of memory fibre formation: *the transition from broad basins to narrow tubes changes the nature of generalization*.

Initial Generalization (1× → 2×) Validation loss improved by 78% (2.97 → 0.65), demonstrating that the attractors being formed are geometrically meaningful. If the model were merely memorizing training sequences statistically, validation performance on duplicated data would not improve—there is no new information in the second copy of eachquestion and answerpair. The improvement indicates that repeated exposure strengthens the manifold structure, making the Q→A trajectories more robust and easier for validation questions to enter.

Precision Refinement (2× → 4×) Continued repetition produced additional validation improvements (70% and 37%), but with a crucial change: the train/val gap collapsed from 1.95 (1-pass) to 0.01 (4-pass). This near-elimination of the gap indicates that the trajectory tubes have become so narrow and well-defined that validation questions flow through them almost as precisely as training questions.

The Geometry of Memory Fibres

The observed training behaviour is inconsistent with classical statistical overfitting but is entirely consistent with the formation of geometric memory fibres. Under a conventional overfitting interpretation, repeated exposure to identical training data would be expected to reduce training loss through memorization while providing no benefit to generalization. Validation loss would plateau or degrade, and the gap between training and validation performance would widen as the model increasingly specializes to the training set. Such a divergence between training and validation curves is a well-established signature of overfitting in statistical learning systems.

By contrast, the behaviour observed in the TBT follows a fundamentally different pattern. Training loss decreases asymptotically with repetition, consistent with the progressive narrowing of tubular attractors rather than rote memorization. Validation loss improves substantially during early repetitions and then closely tracks training loss as repetition increases, indicating that validation trajectories are entering the same geometric structures as training trajectories. Most notably, the train–validation gap collapses toward zero rather than expanding, implying that the learned structures are shared, stable pathways in the reconstructed phase space. This convergence is naturally explained by geometric tube formation and is difficult to reconcile with a purely statistical account of overfitting.

The experimental results match the geometric prediction precisely. The near-zero final gap (0.01) is particularly revealing: it indicates that the manifold contains stable, shared structures—the memory fibres—that both training and validation trajectories follow. These are not memorized sequences but carved geometric channels.

Perplexity and Predictive Confidence

Converting final validation losses to perplexity reveals the progression toward precision:

Repetitions	*Validation Perplexity*	*Interpretation*
1×	$\exp(2.97) \approx 19.5$	High uncertainty, broad distribution
2×	$\exp(0.65) \approx 1.9$	Strong confidence, narrow distribution
3×	$\exp(0.20) \approx 1.2$	Very high confidence, tight distribution
4×	$\exp(0.13) \approx 1.1$	Near-certain prediction, tube-constrained

Table 10: Perplexity progression showing increasing predictive confidence as memory fibres form and narrow.

A perplexity of 1.1 indicates the model is nearly certain about next-token predictions on held-out validation data. This is not statistical certainty—the model has never seen these exact sequences during training—but geometric certainty: the validation questions enter well-defined tubes that lead reliably to their answers.

11.4 Comparison with Corpus Ancora: Domain-Dependent Manifold Topology

The contrasting behaviour between the Solar Systemquestion and answerand Corpus Ancora experiments provides compelling evidence that the TBT learns geometric structures that reflect the intrinsic properties of the training domain.

Structural Comparison

Property	*Corpus Ancora*	*Solar System Q&A*
Domain Type	Compositional, thematic	Factual, point-to-point
Desired behaviour	Creative variation within style	Precise answer retrieval
Manifold Structure	Broad attractor basins	Narrow tubular attractors
Generalization Mode	Transfer to novel phrasings	Precision on exact questions
Training Loss (2×)	0.35	0.16
Validation Loss (2×)	0.65	0.65
Train/Val Gap (2×)	0.30	0.50
Validation Improvement	84% (1× → 2×)	78% (1× → 2×)
Final Gap (4×)	N/A	0.01

Table 11: Comparison of Corpus Ancora and Solar Systemquestion and answerexperiments demonstrating domain-dependent manifold topology.

Geometric Interpretation

Corpus Ancora: Broad Basins Supporting Compositional Generalization

The mythopoetic corpus exhibits thematic recursion, symbolic motifs, and stylistic coherence. This structure creates a manifold with wide attractor basins: many different token sequences can express the same thematic content. When the model encounters a validation sequence with similar thematic structure but different wording, it flows naturally into the same basin. The broad geometry supports creative variation while maintaining semantic and stylistic coherence.

The train/val gap remains moderate (0.30 at 2×) because there is genuine compositional generalization: validation trajectories explore regions of the basin not visited during training, but they remain within the same broad attractor structure.

Solar System Q&A: Narrow Tubes Enforcing Precision

Thequestion and answerdomain demands exact retrieval: "What is the largest planet?" must produce "Jupiter," not "a gas giant" or "something big." This creates a manifold with narrow tubular attractors—memory fibres—that constrain trajectories to specific paths. Eachquestion and answerpair corresponds to a distinct tube connecting the question representation to the answer representation.

The train/val gap collapses to 0.01 (at 4×) because there is no room for variation: the tubes are so narrow that any question falling into a given tube will follow nearly the same trajectory regardless of whether it appeared in training or validation. This is not overfitting but geometric precision.

The Cost of Precision: Limited Generalization to Novel Questions

While the 4-pass model achieves near-perfect precision on questions similar to those in training, this comes at a cost: questions that fall outside the learned tubes cannot be answered reliably. A question like "What is the biggest planet?" (using "biggest" instead of "largest") might fail to enter the correct tube, producing an incorrect or uncertain answer.

This is not a failure of the architecture but an accurate reflection of the task geometry. Factual Q&A, as typically formulated, does not have compositional structure—there is no meaningful sense in which "largest planet" and "biggest planet" share a common attractor unless both phrases appear in training. The manifold learns exactly what it is shown: specific question-answer pathways.

By contrast, attention-based transformers with large-scale pretraining can leverage statistical co-occurrence patterns across billions of tokens to recognize "biggest" and "largest" as synonyms. The TBT, operating on a small domain-specific corpus, has no such statistical backup—it relies purely on geometric structure.

This distinction reveals a fundamental trade-off:

- *Attention-based models*: Broad statistical knowledge, flexible but opaque
- *Takens-based models*: Precise geometric structure, interpretable but domain-specific

11.5 Theoretical Implications: Memory Fibres as Geometric Primitives

The Solar Systemquestion and answerexperiment provides empirical validation for a key theoretical prediction: that different linguistic tasks produce different manifold topologies, and that these topologies can be precisely characterized.

Memory Fibres vs. Attractor Basins

The experimental results suggest that language modelling tasks can be understood in terms of a small number of recurring geometric topologies in semantic phase space. Rather than treating all tasks as variations of a single statistical problem, the TBT reveals that different linguistic objectives induce qualitatively distinct manifold structures. These structures govern how trajectories evolve, how generalization occurs, and how training and validation behaviour should be interpreted. In particular, three recurring topological regimes emerge: broad attractor basins supporting compositional generalization, narrow tubular attractors enforcing precise recall, and mixed topologies that combine both behaviours through networks of constrained pathways.

Attractor Depth and Repetition Dynamics

The progressive narrowing observed across repetitions ($1\times \rightarrow 4\times$) can be understood as a deepening of potential wells in the manifold. Each exposure to aquestion and answerpair:

1. *First Pass*: Discovers approximate location in phase space where question and answer regions connect
2. *Second Pass*: Carves an initial pathway, establishing rough tube geometry
3. *Third Pass*: Narrows the tube by steepening gradients away from the central trajectory
4. *Fourth Pass*: Approaches the precision limit imposed by embedding dimension and discretization

Channel Separation inquestion and answerTasks

Thequestion and answerexperiments used channel separation to distinguish user questions from system answers. Analysis of the training logs reveals interesting channel-specific behaviour:

The "end loss" component, which represents the model's ability to recognize sequence boundaries, shows faster convergence than word prediction loss. By the 4-pass experiment, the model has near-perfect confidence about where answers should terminate (end loss: 0.0017), indicating that sequence boundaries are learned as distinct geometric features of the memory fibres.

This suggests that tube geometry includes not only the Q→A trajectory but also clear demarcation of the answer endpoint. The manifold has learned that certain regions of phase space correspond to "answer complete" states, and trajectories naturally flow toward and stop at these regions.

Table 12: Geometric taxonomy of language modelling tasks in semantic phase space

Topology Type	*Typical Domains*	*Geometric Structure*	*Generalization and Train/Val Behaviour*
Broad Attractor Basins	Natural language generation, creative writing, thematic narrative	Wide basins with smooth gradients toward central attractors; many trajectories converge from diverse entry points	Strong transfer to novel phrasings; moderate train/validation gap reflecting genuine compositional variation
Tubular Attractors (Memory Fibres)	Question–answer systems, fact lookup, deterministic mappings	Narrow tubes with steep gradients away from the pathway; precise geometric constraint	Limited generalization outside learned tubes; train/validation gap collapses toward zero as tubes narrow
Mixed Topologies	Mathematical problem-solving, multi-step reasoning, structured inference	Networks of tubes connected by constrained transitions between intermediate states	Generalization depends on the compositional structure of intermediate steps; train/validation gap varies with pathway complexity

11.6 Practical Implications and Future Directions

The emergence of memory fibres suggests that Takens-based models admit a principled, geometry-aware approach to deployment and training that differs substantially from conventional attention-based systems. Rather than treating all domains as requiring uniform generalization, the geometric structure of the learned manifold indicates that different application regimes benefit from different training strategies.

In precision-critical domains such as medical question answering, legal lookup, or technical documentation, aggressive repetition of training pairs is advantageous. Repetition on the order of three to four passes or more progressively carves deep, narrow memory fibres that enforce deterministic question–answer mappings. In such settings, limited generalization to unseen phrasings is not a defect but a reflection of task requirements, and preprocessing steps that normalize queries into canonical forms can be used to ensure reliable tube entry. The resulting behaviour is highly stable and interpretable, with minimal ambiguity in model outputs.

By contrast, creative and generative domains benefit from the preservation of broad attractor basins rather than the formation of narrow tubes. Minimal repetition, typically one to two passes, allows the manifold to retain wide regions that support compositional generalization and stylistic variation. In this regime, higher perplexity should not be interpreted as model weakness, but as a sign of healthy uncertainty and expressive freedom. Generation proceeds through manifold exploration rather than precise pathway traversal, enabling novel yet coherent outputs.

Mixed domains, such as conversational systems or multi-step reasoning tasks, naturally occupy an intermediate regime. Here, hybrid training strategies are appropriate: high repetition can be applied selectively to factual anchors such as names, dates, or definitions, while reasoning patterns and explanatory text are trained with lower repetition to preserve flexibility. From a geometric perspective, such systems operate on manifolds containing both broad basins and narrow tubes, and an important direction for future work is the development of diagnostics that identify which structures are active during generation.

Beyond domain selection, the results indicate that tube geometry itself can be actively shaped during training. Generalization can be promoted by deliberately widening tubes through data augmentation with paraphrased inputs, the introduction of noise or dropout during training, or early stopping before excessive narrowing occurs. Conversely, precision can be enhanced by aggressively repeating critical sequences, fine-tuning on exact phrasing, or increasing the depth of the manifold projection layers to allow steeper geometric gradients. More complex tasks, such as multi-hop reasoning, may be understood as networks of interconnected tubes, which can be encouraged by designing training sequences that explicitly traverse multiple constrained pathways and by using bridge channels to represent intermediate reasoning states.

The memory fibre framework also offers a natural route to interpretability. For a question such as "What is the largest planet?", a confident prediction corresponds to a trajectory that cleanly enters a narrow tube and follows it stably to the answer state. By contrast, a question like "What is the most massive planet?", if absent from training, may produce a trajectory that approaches but fails to enter the relevant tube, resulting in drift or uncertainty. Quantifying such behaviour through trajectory stability or proximity to known tubes provides a diagnostic signal that is difficult to obtain from attention weights alone. Techniques from dynamical systems analysis, including phase portraits, Lyapunov exponents, and basin stability measures, offer promising tools for formalizing these notions and predicting model behaviour on novel inputs.

Finally, a critical direction for future empirical work is a controlled comparison between Takens-based models and attention-based transformers trained on identical question–answer data under matched repetition protocols. Such experiments would clarify the respective roles of geometric precision and statistical generalization, and help delineate the regimes in which explicit phase-space reconstruction offers decisive advantages over attention-driven architectures.

Predicted Differences:

Property	*Attention-Based*	*Takens-Based*
Loss with Repetition	Training improves, validation plateaus	Both improve initially, then stabilize
Train/Val Gap	Increases with repetition	Decreases to near-zero
Novel Question Performance	Degrades slowly (statistical backup)	Degrades sharply (no tube)
Exact Question Performance	Good but not optimal	Near-perfect after 4×
Interpretability	Attention weights (indirect)	Tube geometry (direct)

Table 13: Predicted behavioural differences between attention-based and Takens-based models onquestion and answer-tasks with progressive repetition.

Such a comparison would clarify when geometric precision is preferable to statistical flexibility and vice versa.

11.7 Limitations and Scope

The results presented here should be interpreted within the constraints of the experimental setting. The Solar Systemquestion and answerexperiments were conducted using a compact vocabulary of approximately 2,000 words within a narrowly defined semantic domain. While this controlled environment is well suited for isolating geometric effects such as memory fibre formation, scaling the approach to substantially larger vocabularies or open-domain question answering is likely to introduce additional dynamics. As the number of distinct question–answer mappings increases, tubular attractors may begin to overlap or interfere, particularly if they occupy nearby regions of semantic phase space. Moreover, a fixed-capacity manifold can only support a finite number of sharply separated tubes, suggesting an eventual trade-off between precision and coverage. Tubes corresponding to widely separated semantic regions may also prove difficult to learn simultaneously, raising questions about sparsity and capacity allocation in large-scale settings. Systematic investigation across larger vocabularies and broader domains will be required to characterize these effects.

A related limitation concerns paraphrase robustness. The experiments reported here relied primarily on exact repetition of question–answer pairs, a design choice intended to expose the formation and refinement of geometric tubes. Introducing paraphrased questions would allow direct measurement of tube width and tolerance to variation, revealing how much linguistic flexibility can be accommodated before a trajectory exits a learned pathway. Such experiments would also clarify whether deliberate data augmentation can be used to widen tubes in a controlled manner, or whether widening necessarily trades away the precision observed in the repeated-exposure regime. Preliminary observations suggest that memory fibres are narrow, with even minor rephrasings often failing to enter the correct tube, but these impressions remain anecdotal without targeted experimentation.

Finally, it is important to situate the memory fibre approach in relation to existing question-answering paradigms, particularly retrieval-augmented generation (RAG). Whereas RAG systems retrieve external documents and then generate responses using a language model, the Takens-based approach encodes question–answer mappings directly into the geometry of the learned manifold. This yields fast, deterministic, and interpretable behaviour for questions that fall within established tubes, but offers limited flexibility when confronted with genuinely novel queries. RAG systems, by contrast, are more adaptable and capable of handling unseen questions, but at the cost of additional infrastructure, latency, and reduced interpretability. These approaches should therefore be viewed as complementary rather than competing. A promising direction for future work is the development of hybrid systems that rely on memory fibres for frequent, well-defined queries, while deferring to retrieval-based mechanisms when geometric confidence is low or tube entry fails.

11.8 Summary

The Solar Systemquestion and answerexperiment demonstrates that the TBT can learn precise factual mappings through the formation of memory fibres: narrow, tubular attractors that connect questions to answers with minimal deviation. As identical question–answer pairs are progressively repeated during training, the geometry of the learned manifold undergoes systematic refinement. The initial increase in repetition (from $1\times$ to $2\times$) produces a rapid reduction in both training and validation loss (85% and 78%, respectively), marking the emergence of stable Q$\rightarrow$A pathways in phase space. Further repetition (from $2\times$ to $4\times$) does not introduce qualitatively new structures but instead sharpens existing ones, leading to asymptotic improvements in precision as trajectories become increasingly constrained within narrow tubes.

This geometric interpretation is reinforced by the behaviour of the train–validation gap, which collapses toward zero as repetition increases, reaching 0.01 in the $4\times$ condition. Such convergence indicates that training and validation trajectories are flowing through the same shared geometric structures rather than diverging through memorization. The final validation perplexity of approximately 1.1 reflects near-certain next-token predictions once a trajectory has entered a well-formed tube, consistent with deterministic behaviour within a constrained manifold region.

The contrast with the Corpus Ancora experiment further highlights the task-dependent nature of the learned topology. Compositional domains characterized by thematic variation induce broad attractor basins that support strong generalization across novel phrasings, whereas factual retrieval tasks induce narrow tubular structures that favor high precision at the expense of transfer. Together, these results indicate that the TBT does not impose a fixed representational geometry, but instead adapts its manifold structure to the intrinsic demands of the task.

This is not a limitation but a feature: the architecture learns the geometric properties inherent to the task. The cost of precision is reduced robustness to novel phrasings, but the benefit is interpretable, deterministic, and verifiable behaviour on known questions.

The memory fibre phenomenon validates the core theoretical framework: that language can be modelled as trajectories through learned semantic manifolds, and that different linguistic tasks produce different manifold topologies. The TBT does not impose a single geometric structure on all language—it discovers and refines the structure that exists in the training data.

These results establish memory fibres as a fundamental geometric primitive for understanding how language models encode factual knowledge, complementing the broad attractor basins that support compositional generalization. Together, these structures suggest a geometric taxonomy of linguistic competence that may guide future architecture design and training protocols.

12 Corpus Ancora Generative Model: Long-Form Mythopoetic Structure and Semantic Manifold Dynamics

12.1 Motivation and Dataset Rationale

The third experiment explores MARINA's behaviour in a high-level generative regime using an engineered mythopoetic corpus derived from the Corpus Ancora. This text is intentionally constructed to exhibit strong stylistic coherence, recurring symbolic motifs, layered metaphors, and long-form conceptual arcs. Unlike the Brown Corpus, which reflects natural linguistic distributions, or the Solar System dataset, which is purely factual and task-bound, the Corpus Ancora serves as a probe into MARINA's ability to model deep semantic structure and aesthetic continuity.

This setting pushes the architecture to engage with narrative curvature, thematic attractors, and stylistic basins in the learned manifold. The text's recursive motifs and symbolic vocabulary create distinctive geometric features: trajectories must return to conceptual regions that represent mythic elements, while simultaneously evolving forward in narrative time. This allows us to observe whether the delay-coordinate reconstruction mechanism naturally supports long-range coherence, rhythmic structure, and conceptual recurrence.

Furthermore, generative modelling of curated mythos material tests the architecture's ability to synthesize new text that is not merely grammatically valid but semantically and stylistically aligned with the training corpus. Success here would indicate that the manifold learned by MARINA supports creative generative flows, not only factual reasoning and statistical prediction.

In this context, the Corpus Ancora experiment provides insight into the expressive range of the TBT: whether its geometric structure can carry the weight of artistic style, symbolic structure, and mythic narrative form.

12.2 Experimental Design

The Corpus Ancora was processed into 444 training sequences of 256 tokens each using sliding windows with 50% overlap (stride of 128 tokens). The complete corpus vocabulary comprised 7,748 unique words. The model architecture remained identical to previous experiments: 2.7M parameters with exponential delay embedding [1, 2, 4, 8, 16, 32] and three-channel topology (user, system, bridge).

Two experiments were conducted to probe the nature of learning in the Takens-based architecture:

Experiment 1 (Single-Pass): Standard training on the corpus for 50 epochs

Training samples: 400 sequences Validation samples: 44 sequences Total exposure: 20,000 sequence presentations

Experiment 2 (Doubled-Data): Training on the corpus concatenated with itself

Training samples: 801 sequences (400 unique + 400 duplicates) Validation samples: 88 sequences (44 unique + 44 duplicates) Total exposure: 40,050 sequence presentations Critical detail: Every training sequence appears exactly twice; validation sequences are also duplicated

The doubled-data experiment was designed to test a specific hypothesis: if MARINA learns geometric structure (manifold topology, attractor basins, flow dynamics) rather than statistical patterns, then repeated exposure to identical sequences should strengthen the learned geometry and improve generalization, rather than inducing overfitting.

12.3 Results: Single-Pass Training

The single-pass experiment demonstrated strong convergence and stable learning dynamics: *Key Observations:*

Table 14: Training and validation performance metrics

Metric	Initial (Epoch 1)	Final (Epoch 50)	Summary
Training Loss	8.09	1.68	Total reduction of 6.41 (79% improvement); average decrease of 0.128 per epoch
Validation Loss	7.20	3.99	Total reduction of 3.21 (45% improvement); final perplexity $\exp(3.99) \approx 54$

Monotonic Improvement: Both training and validation loss decreased steadily throughout all 50 epochs with no plateau, suggesting the model had not yet saturated the geometric structure of the corpus.

No Catastrophic Overfitting: Despite the small dataset (400 training samples), validation loss tracked training loss proportionally without divergence. The train/val gap at epoch 50 was 2.31, indicating reasonable generalization.

Three-Phase Learning Pattern:

Epochs 1-10: Rapid discovery ($\sim$0.40 loss/epoch) - finding major attractors Epochs 10-25: Refinement ($\sim$0.18 loss/epoch) - sharpening basin boundaries Epochs 25-50: Polishing ($\sim$0.06 loss/epoch) - fine-tuning manifold curvature

This pattern is characteristic of geometric learning: the model first discovers the coarse structure of the semantic manifold, then progressively refines its representation of that structure.

12.4 Results: Doubled-Data Experiment

The doubled-data experiment produced results that are difficult to reconcile with statistical learning theory but entirely consistent with the geometric interpretation of TBTs.

12.5 Theoretical Significance: Evidence of Geometric Learning

The doubled-data results provide compelling empirical evidence that the TBT learns fundamentally differently from statistical language models.

Table 15: Training and validation performance metrics

Metric	**Initial (Epoch 1)**	**Final (Epoch 50)**	**Summary**
Training Loss	7.37	0.35	Total reduction of 7.02 (95% improvement); average decrease of 0.140 per epoch
Validation Loss	6.29	0.65	Total reduction of 5.64 (90% improvement); final perplexity $\exp(0.65) \approx 1.9$

Table 16: Critical comparison between single-pass and doubled-data training

Metric	**Single-Pass**	**Doubled-Data**	**Change**
Final Training Loss	1.68	0.35	-79%
Final Validation Loss	3.99	0.65	-84%
Train/Validation Gap	2.31	0.30	-87%
Final Perplexity	54.0	1.9	-96%

The Paradox of Improved Generalization

Under standard statistical learning theory, exposing a model to duplicated training data should not improve validation performance. The validation set contains sequences the model has never seen during training; duplicating the training data provides no new information about these held-out sequences. Yet validation loss improved by 84% (3.99 → 0.65).

Three explanations are theoretically possible:

1. *Statistical overfitting with spurious validation improvement:* The model memorized training sequences so thoroughly that it incidentally improved on validation data through some artifact. However, this is contradicted by the reduced train/val gap (2.31 → 0.30), which indicates less overfitting, not more.
2. *Insufficient training in single-pass experiment:* Perhaps the single-pass model simply hadn't converged, and the doubled-data experiment merely provided more training iterations. However, both experiments ran for 50 epochs, and the single-pass validation loss was still improving at epoch 50, suggesting continued learning capacity in both cases.
3. *Geometric structure learning:* The model learned topological invariants—attractor basins, flow patterns, manifold curvature, thematic structure—that exist in the corpus regardless of how many times sequences are presented. Repeated exposure strengthened these geometric features, making the manifold more robust and causing trajectories to flow more predictably even on unseen validation sequences.

Only the third explanation accounts for all observed phenomena: the dramatic validation improvement, the reduced overfitting gap, the continued learning at epoch 50, and the characteristic three-phase learning pattern.

Attractor Reinforcement Dynamics

The learning curves reveal distinct convergence behaviour between single-pass and doubled-data experiments (see Figure X). In the doubled-data experiment, the model reaches loss values in epoch 20 (training: 1.34, validation: 1.65) that the single-pass experiment never achieves even at epoch 50. This acceleration cannot be attributed to seeing "more data"—the sequences are identical.

The geometric interpretation provides a natural explanation: the first exposure to each sequence allows the model to discover the approximate location of attractors in semantic phase space. The second exposure refines these attractors, deepening their basins and sharpening their boundaries. This creates a more stable manifold topology in which trajectories flow more predictably toward their target attractors, improving both training performance and—critically—generalization to unseen sequences that share the same underlying geometric structure.

Channel Separation and Sequence Boundaries

The model's treatment of sequence boundaries provides additional evidence of geometric refinement. The "end loss" component, which represents the model's ability to recognize sequence termination, showed marked improvement in the doubled-data experiment:

Single-pass final end loss: 0.0036 Doubled-data final end loss: 0.0010

This improvement suggests that sequence boundaries are learned as geometric features of the manifold—specific regions where trajectories naturally terminate—rather than as statistical markers. The reinforced geometry makes these boundary regions more distinct and recognizable.

Implications for the Takens Framework

These results strengthen the central theoretical claim of this work: that language models based on explicit delay-coordinate reconstruction learn qualitatively different representations than attention-based models.

An attention-based transformer stores and retrieves patterns through learned similarity functions over token embeddings. When presented with duplicated training data, such a model would strengthen the same statistical associations it learned on the first pass, potentially improving training loss through better memorization but offering no mechanism for improved generalization on held-out data.

A TBT, by contrast, reconstructs the geometry of semantic phase space from temporal observations. The manifold it learns is not a collection of memorized sequences but a continuous structure with attractors, repellers, and flow dynamics. Strengthening this structure through repeated observation makes the entire manifold more coherent, benefiting all trajectories that traverse it—including those from validation sequences.

This distinction is not merely conceptual: the 84% improvement in validation loss on duplicated data provides quantitative evidence that the learning mechanism is fundamentally geometric rather than statistical.

12.6 Generative behaviour and Stylistic Coherence

[Note: This section would contain examples of generated text from both models, demonstrating thematic coherence and stylistic consistency. Generated samples should be included here once inference examples are available.]

12.7 Discussion: The Geometry of Mythopoetic Language

The Corpus Ancora experiment reveals that structured, thematically coherent text creates distinctive manifold geometry that a Takens-based architecture can learn and exploit. The mythopoetic corpus, with its recursive motifs, symbolic structures, and long-range narrative arcs, appears to produce a semantic manifold with well-defined attractors and clear flow dynamics.

This has several implications:

1. *Domain-Specific Manifold Structure:* Different types of language may produce manifolds with different geometric properties. Mythopoetic text, with its intentional thematic recursion, creates deeper, more stable attractors than generic natural language. This suggests that the Takens framework may be particularly well-suited for modeling structured genres: poetry, liturgical text, legal language, technical documentation—any domain where semantic patterns exhibit strong regularities.
2. *Training Efficiency for Coherent Corpora:* The doubled-data experiment suggests that when a corpus has strong geometric structure, repeated exposure is not wasted but actively beneficial. This contradicts conventional wisdom in deep learning, which holds that data augmentation should introduce variation rather than repetition. For Takens-based models, repetition strengthens geometry.
3. *Generalization Through Geometric Invariants:* The validation improvement demonstrates that models can generalize by learning structural properties that transcend specific token sequences. A validation sequence the model has never seen can nonetheless flow naturally through the learned manifold if it shares the same thematic structure, stylistic constraints, and symbolic vocabulary as the training corpus.

12.8 Limitations and Future Directions

While the Corpus Ancora experiment provides strong evidence for geometric learning, several limitations must be acknowledged:

Corpus Engineering: The Corpus Ancora was deliberately constructed to exhibit coherent structure. It remains to be seen whether similar geometric benefits emerge with less structured corpora, or whether the doubled-data effect is specific to texts with strong thematic organization.

Scale Questions: These experiments used a small model (2.7M parameters) and a compact corpus (7,748 words, 444 sequences). Scaling to larger vocabularies, longer sequences, and deeper architectures may reveal different dynamics.

Comparison with Attention-Based Models: A direct comparison experiment—training an attention-based transformer on the same corpus with the same doubled-data protocol—would provide stronger evidence for the uniqueness of the geometric learning pattern. Such experiments are planned for future work.

Interpretability of Learned Attractors: While we can infer the existence of attractor structures from loss dynamics, we have not yet developed methods to visualize or characterize these attractors directly. Techniques from dynamical systems theory—phase portraits, Lyapunov exponents, basin stability analysis—may provide tools for such investigations.

12.9 Summary

The Corpus Ancora experiment demonstrates that the TBT can model long-form, stylistically coherent text with strong thematic structure, and that repeated exposure to identical training sequences improves generalization rather than degrading it. This behaviour is inconsistent with classical statistical learning expectations but follows naturally from a geometric learning interpretation, in which repeated exposure refines the underlying manifold rather than memorizing surface patterns. The results suggest that the learned manifold encodes structural invariants that transcend specific token sequences, allowing the model to capture recurring motifs, symbolic recursion, and stylistic coherence characteristic of mythopoetic language. Such material induces distinctive attractor geometry—broad basins shaped by thematic regularities—which the delay-coordinate reconstruction framework is able to represent effectively. Together, these findings provide empirical support for the core theoretical claim of this work: that language modeling can be understood as the evolution of trajectories through learned semantic manifolds, and that explicit Takens-style delay embedding offers a principled mechanism for reconstructing those manifolds. The dramatic improvement in validation performance observed in the doubled-data experiment, corresponding to an 84% reduction in loss, provides quantitative evidence that this architecture learns a form of structure that is not merely an approximation of attention-based transformers, but a geometrically distinct mode of representation.

13 Discussion

13.1 What This Work Demonstrates

The results presented in this paper show that a Takens-based interpretation of language modelling is not merely a theoretical curiosity, but a practical and operational framework. In the earlier paper, Pairwise Phase Space Embedding in Transformer Architectures, the central claim was that attention performs a form of implicit delay-coordinate reconstruction. The argument was mathematical and structural: attention layers compute relational projections that resemble Takens-style embeddings of a temporal signal. What remained untested was whether a model built explicitly upon this idea—one that abandons attention entirely—could function as a viable language model.

The present work closes that loop. By constructing a transformer whose sole mechanism for context integration is explicit, exponentially spaced delay sampling, we demonstrate that phase-space reconstruction is sufficient to learn broad linguistic patterns, factual relations, and stylistically coherent generative structure. The architecture described here does not approximate attention; it replaces it with a different mathematical principle. The fact that such a model converges stably, generates coherent text, and performs structured reasoning suggests that the core insight of the earlier paper was correct: language models do not require quadratic relational search if they can instead reconstruct the dynamical state of the sequence.

In this sense, the TBT represents more than an alternative design. It serves as a proof that the internal logic of modern language models can be reframed in geometric and dynamical terms. The notion that meaning resides in trajectories—rather than in static semantic vectors—gains empirical grounding when the architecture built around this notion behaves coherently across multiple linguistic domains. The results do not claim that this dynamical approach is superior to attention-based methods, but they do show that the machinery needed for language understanding can be simpler, more interpretable, and more closely aligned with established mathematical theory.

By demonstrating that a functional language model can be constructed around explicit phase-space reconstruction, this work provides both validation of the prior theoretical proposal and a new path forward for understanding how sequence models operate. It suggests that the success of transformers may arise not primarily from attention as a mechanism of relevance retrieval, but from the implicit embedding of temporal dynamics in high-dimensional space. The TBT makes that embedding explicit, thereby offering a clearer window into the geometry of linguistic behaviour and the mechanisms by which models synthesise meaning from sequences.

13.2 What This Work Does Not Claim

While strengthening the case for a dynamical interpretation of language modelling, this work makes several important disclaimers. First, it does not claim that Takens-based architectures outperform attention-based transformers, nor that phase-space reconstruction is the exclusive or optimal foundation for all future language models. The purpose is more fundamental: to demonstrate that explicit reconstruction of semantic trajectories is sufficient for coherent linguistic behaviour, showing that attention is not the only viable means of modelling context.

Second, the work does not exhaust the design space of dynamical architectures. The models presented are intentionally modest in scale and computational footprint, trained on constrained datasets under resource-limited conditions to ensure reproducibility and conceptual clarity. The architecture is a first implementation; deeper manifold projections, richer channel structures, and alternative delay schedules remain unexplored. The scalability of these models is an open question.

Third, it is not asserted that the representations learned by MARINA correspond to human linguistic intuitions or cognitive processes. Although the dynamical framing resonates with theories of temporal cognition, the architecture should be understood as a mathematical construct, not a model of human thought.

Finally, this work does not resolve all interpretability questions. While the Takens-based architecture is more transparent—reconstructing state directly from observable delays—its internal manifold remains a complex learned object. Full characterisation of its geometry lies beyond this paper's scope.

These limitations emphasise that the contribution lies not in claiming superiority, but in demonstrating the viability of a different mathematical lens and illuminating language model behaviour in a way that complements the attention paradigm.

13.3 Implications for Transformer Theory

The development of a TBT provides an opportunity to reconsider the foundations of transformer models more broadly. Attention has long been treated as the defining feature of the transformer architecture—the mechanism responsible for contextual integration, relational reasoning, and long-range coherence. Yet the success of MARINA suggests that the true power of transformers may lie not in attention as a specific computational primitive, but in the broader capacity of deep models to embed and evolve trajectories in high-dimensional space.

From this perspective, attention becomes one possible method for approximating a deeper operation: reconstructing a latent semantic state from an observable sequence. In traditional transformers, the attention matrix implicitly performs a form of multi-scale, relational sampling of the past. By contrast, the TBT makes the reconstruction explicit. It does not search across past tokens for relevance; it assumes relevance is encoded in the geometric structure of the trajectory itself. The fact that both architectures can produce coherent linguistic behaviour suggests that transformers may be understood more fruitfully as devices for state estimation and dynamical evolution rather than tools for relational lookup.

This reframing carries important theoretical implications. If a transformer functions by inferring and updating a position on a learned manifold of linguistic states, then the meaning of a token is not determined by its embedding alone, but by how it perturbs the system's trajectory. This view aligns naturally with the dynamical systems community, where attractors, flows, and curvature define behaviour. Under this lens, attention heads may be interpreted as learned estimators of the dynamical coordinates needed to reconstruct the current state. Their multiplicity, diversity, and instability across layers may reflect not fragmentation of linguistic knowledge, but an attempt to approximate a manifold that a Takens-based model constructs in a single, coherent pass.

Moreover, this viewpoint helps illuminate certain empirical phenomena observed in large transformers. Behaviours such as in-context learning, emergent reasoning, or spontaneous structure formation become less mysterious when understood as consequences of how trajectories evolve in a learned dynamical landscape. The model is not retrieving facts or performing symbolic operations; it is navigating regions of the manifold whose geometry encodes patterns of language, reasoning pathways, and even stylistic conventions. A Takens-based architecture, by design, foregrounds this geometry and demonstrates that linguistic competence emerges from the evolution of state, not from pairwise relevance computation.

Seen in this light, the TBT is not merely a simplification of the attention mechanism—it is a conceptual clarification. It shows that transformers work because they maintain and update a latent state embedded in a structured manifold, and that attention is one way (but not the only way) to construct the coordinates of that state. The explicit reconstruction performed by MARINA reveals what attention may have been implicitly doing all along: learning how to embed a temporal signal into a coherent phase space on which linguistic meaning becomes a geometric property.

13.4 Implications for Architecture Design

Recasting language modelling as a problem of dynamical reconstruction rather than relational attention invites a reconsideration of how sequence architectures should be designed. Traditional transformers assume that effective modelling requires a mechanism for every token to attend to every other token, expanding memory consumption quadratically and embedding context retrieval directly into the computational graph. MARINA, by contrast, demonstrates that a model can maintain contextual fidelity by reconstructing a latent dynamical state through structured delay embeddings, without tracking or comparing every pair of tokens. This shift in perspective has several architectural consequences.

First, memory becomes a function of state representation rather than history storage. In a Takens-based system, the capacity to hold context is defined by the dimension of the reconstructed manifold and the richness of its learned geometry. Context is encoded implicitly in the state, not preserved explicitly in key–value caches or large attention tensors. This leads to architectures whose memory footprint is fixed regardless of sequence length, allowing the model to process arbitrarily long inputs without expanding internal buffers. Such behaviour aligns more closely with classical dynamical systems, where the state evolves continuously and does not require storing the entire trajectory.

Second, the architecture foregrounds the design of temporal sampling strategies. Because the model explicitly chooses which delays form its state representation, the exponential spacing of delays becomes a dial for shaping linguistic sensitivity. Fine-grained delays capture local syntactic transitions, while larger delays capture clause-level or narrative structure. The architecture therefore encourages thinking in terms of temporal scales rather than attention patterns. This poses a different design challenge: instead of learning how to search the past, the model must learn how to evolve from a reconstructed present.

Third, the incorporation of channel topology introduces architectural possibilities that attention-based models do not easily support. By separating user inputs, internal reasoning, and outputs into distinct manifold regions, the model enforces structural identity boundaries at the geometric level. This enables forms of modularity, role separation, and controlled reasoning that are difficult to achieve with prompt engineering or soft supervision alone. Such structures may offer pathways toward more interpretable or controllable language models, where internal transformations are both explicit and constrained.

Finally, the TBT encourages a shift from designing architectures around data flow to designing them around geometric flow. The central question becomes not how information is passed between layers, but how trajectories evolve across layers. The manifold projection, temporal mixing, and update dynamics collectively determine the curvature and attractors of the semantic space. This perspective reframes architecture design as the sculpting of a dynamical landscape—one in which stability, expressiveness, and generalisation arise from the properties of the flow rather than the combinatorics of attention.

Taken together, these implications suggest that future architectures may benefit from incorporating explicit geometric principles into their design. By revealing how language modelling can emerge from phase-space reconstruction, MARINA invites a broader exploration of architectures that prioritise dynamical coherence, temporal structure, and geometric interpretability over relational search. This does not replace attention as a useful tool, but it expands the space of viable mechanisms and offers a new foundation for reflecting on why transformer-style models work at all.

13.5 Limitations and Open Questions

Although the results presented here establish the viability of Takens-based language modelling, several limitations must be acknowledged. These limitations do not undermine the conceptual contribution; rather, they clarify the scope of the present work and highlight directions for future research. As with any first implementation of a new architectural paradigm, the boundaries encountered here should be viewed as guideposts rather than shortcomings.

The most immediate limitation is one of scale. The models trained in this study are intentionally modest in parameter count and trained on constrained datasets to emphasise interpretability, reproducibility, and conceptual clarity. This leaves open the question of how the architecture behaves at the scale of modern large language models, where dimensionality, dataset diversity, and training dynamics differ substantially. It remains to be seen how the manifold geometry evolves under billions of parameters or whether deeper projection layers and larger temporal windows would lead to emergent behaviours analogous to those observed in attention-based systems.

Another limitation lies in the restricted exploration of delay schedules and projection depths. The exponential delay scheme used here is mathematically motivated and has desirable multi-scale properties, but it is not the only possible temporal sampling strategy. Different domains may require different delay structures, and it is unknown how sensitive performance is to the granularity or distribution of delays. Similarly, the manifold projection layer used in this work is deliberately simple. More expressive projections—nonlinear, recurrent, hierarchical—may yield richer geometric structures that better capture long-range dependencies or higher-order semantics.

The interpretation of the learned manifold also remains a largely open question. While the Takens-based framework is more transparent in its principles than attention, the geometry of the learned state space is still a high-dimensional object. Characterising its curvature, attractors, stability properties, and phase transitions will require tools from dynamical systems theory that go beyond the scope of this paper. Understanding how linguistic categories, reasoning patterns, or stylistic features manifest as regions or flows within the manifold poses both a challenge and an opportunity for interdisciplinary research.

Finally, the experiments in this study focus on domains where ground truth is easily defined or stylistic coherence is directly observable. More complex tasks—multi-step reasoning, dialogue coherence, abstract question answering—will require a deeper understanding of how channel topology, state transitions, and manifold structure interact. It is an open question whether explicit phase-space reconstruction offers advantages in such settings or whether hybrid architectures that combine delay embeddings with selective attention mechanisms may prove most effective.

These limitations should not be viewed as obstacles but as invitations. They mark the frontier of a research trajectory that begins with the central insight of the earlier theoretical work: that language modelling can be understood as the reconstruction and evolution of a latent dynamical state. The TBT presented here is a first step in operationalising that insight. The open questions it raises point toward a rich and largely unexplored landscape where geometry, dynamics, and linguistic structure converge.

14 Conclusion and Future Directions

This work set out to test a simple but far-reaching idea: that the internal mechanisms enabling language models to function may be more naturally understood as dynamical reconstruction rather than as relational attention. In the earlier paper, Pairwise Phase Space Embedding in Transformer Architectures, this idea appeared as a conceptual lens—a reinterpretation suggesting that attention layers implicitly approximate the delay-coordinate embeddings used in nonlinear dynamical systems. The contribution remained theoretical. What was unknown was whether this view could be made operational: whether a model built explicitly upon Takens-style reconstruction could learn language at all.

The TBT presented in this study answers that question affirmatively. By replacing attention entirely with an explicit, exponentially sampled delay embedding, and by projecting this reconstruction onto a learned semantic manifold, we demonstrate that a transformer can achieve stable training, coherent text generation, factual reasoning, and stylistically constrained creativity without relying on quadratic relational search. The three experimental models—spanning general-domain language, structured question–answering, and mythopoetic generation—show that the architecture is both flexible and expressive, even at modest scale.

This result does not diminish the practical achievements of attention-based transformers; rather, it clarifies the underlying principle that may unify them. Whether implemented implicitly through attention or explicitly through delay embeddings, the essential operation appears to be the reconstruction of a latent state from a temporal signal. Language models succeed not because they perform exhaustive relevance lookup, but because they learn a geometry in which meaning evolves as a trajectory. Seen from this perspective, the TBT does not oppose the traditional architecture but reveals a deeper structure that may have guided its success.

The broader implication is that language modelling may not require the architectural complexity that current practice assumes. If coherent linguistic behaviour can emerge from fixed-size delay buffers, topologically separated channels, and lightweight manifold projections, then the design space for future models is far wider than the attention paradigm alone suggests. Such models may offer advantages in interpretability, computational efficiency, or controllability, particularly in contexts where identity separation or constrained reasoning pathways are desirable.

At the same time, this work represents only a beginning. Many questions remain open: how these models behave at scale, how their manifolds evolve with increasing data, how delay schedules interact with linguistic structure, and whether hybrid architectures combining attention and dynamical reconstruction may provide additional benefits. The answers to these questions will require further exploration, deeper mathematical analysis, and broader experimentation.

What this paper establishes is that the dynamical perspective is not speculative. It works. It provides coherent predictions, aligns with empirical behaviour, and opens a new line of inquiry into the nature of meaning, context, and computation in language models. The hope is that this work will serve as both a technical foundation and an invitation—to reconsider the mechanisms we take for granted, to explore alternatives grounded in established mathematical theory, and to broaden our understanding of how language can be modelled by machines.

14.1 Final Thoughts

The transformer architecture's success is undeniable, but it need not be the final word. If attention can be viewed as implicit delay embedding, then perhaps our explicit formulation reveals something fundamental about what language models are actually doing. Whether this specific architecture succeeds or fails at scale, the theoretical perspective it embodies—language as dynamical trajectory, context as phase space reconstruction, meaning as geometry—may prove valuable in understanding the next generation of models.

We offer MARINA not as a finished product but as a proof-of-concept and an invitation: there are other ways to think about language, other lenses through which to view the problem, and other paths that might lead somewhere interesting. This is one such path. We hope others will explore it further.

Acknowledgments

This work was conducted independently without institutional support, grant funding, or access to computational clusters. We thank the open-source community for tools (PyTorch, NumPy, Matplotlib) that made this research possible on consumer hardware. We thank the creators of the Brown Corpus for maintaining a freely available benchmark dataset suitable for small-scale research.

References

[1] K. R. Haylett. *Finite Tractus: The Hidden Geometry of Language and Thought*. ISBN-13:979-8281127776, April 2025.

[2] K. R. Haylett. Pairwise Phase Space Embedding in Transformer Architectures. `www.finitemechanics.com`, May 2025.

[3] M. Ostrow, A. Eisen, and I. Fiete. Delay embedding theory of neural sequence models. *arXiv preprint arXiv:2406.11993*, 2024.

[4] F. Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence*, Springer Lecture Notes in Mathematics, 898:366–381, 1981.

[5] A. Vaswani, N. Shazeer, N. Parmar, et al. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[6] A. Gu, K. Goel, and C. Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

[7] A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv:2312.00752*, 2023.

[8] B. Peng et al. RWKV: Reinventing RNNs for the transformer era. *arXiv:2305.13048*, 2023.

[9] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, 2020.

[10] M. Zaheer et al. Big Bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, 2020.

[11] Y. Tay et al. Efficient transformers: A survey. *arXiv:2009.06732*, 2020.

[12] Brown Corpus. A standard corpus of present-day edited American English. Francis and Kucera, Brown University, 1961.

Appendix A - Architectural Details

Complete Model Specification

Table 17: Takens Embedding Architecture Summary

Type	Component	Purpose	Input	Output	Key Params
Core Module	TakensEmbedding	Delay embeddings via Takens' theorem	(B, L, D)	$(B, L, M+1, D)$	`dim`, `delays`
	AdaptiveTakens	Learnable with adaptive weighting	(B, L, D)	(B, L, D_{out})	`dim`, `dropout`
Utility	`exp_delays()`	Exponential delays [1,2,4,8,...]	`max`	List[int]	`base`
	`log_delays()`	Logarithmic delays	`max`	List[int]	`num`
Key Features					
Default delays: $[1, 2, 4, 8, 16, 32, 64, 128]$ Embedding: $[x(t), x(t-\tau_1), \ldots, x(t-\tau_m)]$ Output dim: $(M+1) \times D$ Adaptive: Projection + LayerNorm					

B=batch, L=seq len, D=emb dim, M=delays, D_{out}=output dim

A Explicit Construction of Takens Delay Embeddings and Adaptive Manifold Projection

A.1 Purpose and Scope

Sections 4.6–4.9 introduce exponential delay embeddings and adaptive manifold projection as the mechanism by which the Takens-Based Transformer reconstructs semantic phase space from a token stream. Those sections focus on theoretical motivation and architectural implications; full variable declarations and implementation-level detail are necessarily condensed.

This appendix provides a complete and explicit account of the mathematics underlying that mechanism. In particular, it:

1. defines all variables and indices unambiguously,
2. describes the construction of the Takens delay-coordinate embedding,
3. explains the learned projection onto the semantic manifold,
4. clarifies the role of the bias term as *reference positioning*, and
5. provides a worked example illustrating the full reconstruction pipeline.

The intent is that a reader encountering the manifold projection equations in the main text can refer here to see exactly how those quantities are computed and interpreted.

A.2 Notation and Indexing Conventions

We adopt the following conventions throughout.

Dimensions

- $B \in \mathbb{N}$: batch size
- $L \in \mathbb{N}$: sequence length
- $d \in \mathbb{N}$: token embedding dimension
- $M \in \mathbb{N}$: number of delay offsets
- $d_{\text{out}} \in \mathbb{N}$: manifold (model) dimension

Indices

- $b \in \{1, \ldots, B\}$: batch index
- $t \in \{0, \ldots, L-1\}$: token (time) index
- $i \in \{1, \ldots, M\}$: delay index

Uppercase symbols denote dimensional quantities; lowercase symbols denote indices into those dimensions.

A.3 Input Representation

The input to the Takens embedding module is a batch of embedded token sequences:

$$\mathbf{X} \in \mathbb{R}^{B \times L \times d},$$

with individual token embeddings

$$\mathbf{x}_{b,t} \in \mathbb{R}^{d}.$$

Each sequence is treated as a single observable time series, consistent with the assumptions of Takens' Delay Embedding Theorem.

A.4 Delay-Coordinate Reconstruction

A.4.1 Delay Schedule

We define a fixed set of integer delay offsets

$$\{\tau_1, \tau_2, \dots, \tau_M\}, \qquad \tau_i \in \mathbb{N}.$$

In the implementation used throughout this work, delays are exponentially spaced:

$$\tau_i = 2^{i-1}.$$

This logarithmic spacing captures linguistic structure across multiple temporal scales while maintaining a fixed memory footprint independent of sequence length.

A.4.2 Raw Delay-Coordinate Vector

For each batch element b and time step t, we construct the raw delay-coordinate vector

$$\tilde{\mathbf{z}}_{b,t} = \Big[\mathbf{x}_{b,t}, \mathbf{x}_{b,t-\tau_1}, \mathbf{x}_{b,t-\tau_2}, \dots, \mathbf{x}_{b,t-\tau_M}\Big] \in \mathbb{R}^{(M+1)d}.$$

If $t < \tau_i$, the corresponding delayed embedding is replaced with a zero vector. This zero-padding ensures that the delay embedding is well-defined at all time steps.

The vector $\tilde{\mathbf{z}}_{b,t}$ is the explicit Takens delay embedding: it represents a local reconstruction of the system's phase-space neighborhood using only past observations of the token stream.

A.5 Adaptive Manifold Projection

The delay-coordinate vector is high-dimensional and sparse. It is therefore mapped onto a dense latent manifold where semantic trajectories evolve smoothly. We distinguish three successive representations.

A.5.1 Linear Projection (Pre-Normalization State)

The raw delay vector is passed through a learned affine projection:

$$\mathbf{h}_{b,t} = \mathbf{W}_p\, \tilde{\mathbf{z}}_{b,t} + \mathbf{b}_p,$$

where

- $\mathbf{W}_p \in \mathbb{R}^{d_{\text{out}} \times (M+1)d}$ is the projection matrix,
- $\mathbf{b}_p \in \mathbb{R}^{d_{\text{out}}}$ is the bias vector, and
- $\mathbf{h}_{b,t} \in \mathbb{R}^{d_{\text{out}}}$.

The matrix $\mathbf{W}_p$ learns how information from different temporal scales should be weighted and combined. No pairwise comparison or retrieval is performed; all contextual information enters exclusively through the explicit delay coordinates.

A.5.2 Role of the Bias Term (Reference Positioning)

The bias vector $\mathbf{b}_p$ plays no role in temporal reconstruction and is not required by Takens' theorem. Its function is purely geometric.

Specifically, the bias provides *reference positioning* of the learned manifold within the projection space. It implements a translation that determines where the manifold sits relative to the coordinate origin, without altering its intrinsic geometry, curvature, or topology.

This separation is deliberate:

- $\mathbf{W}_p$ determines orientation, scaling, and curvature;
- $\mathbf{b}_p$ determines reference placement.

The bias introduces no temporal structure, no memory, and no contextual information. It serves only to decouple manifold placement from padding artifacts and early-sequence effects.

A.5.3 Normalized Manifold State

The projected vector is then normalized:

$$\mathbf{z}_{b,t} = \mathrm{LayerNorm}(\mathbf{h}_{b,t})\,.$$

The vector $\mathbf{z}_{b,t}$ is the final reconstructed phase-space state. It represents the model's current position on the learned semantic manifold and is the quantity passed to temporal mixing layers, channel-aware decoding heads, and next-token prediction.

A.5.4 Interpretation of the State Chain

The roles of the three vectors are distinct:

- $\tilde{\mathbf{z}}_{b,t}$: explicit history (where the system has been),
- $\mathbf{h}_{b,t}$: learned geometric combination of that history,
- $\mathbf{z}_{b,t}$: current manifold state.

Temporal structure is explicit; geometric structure is learned.

A.6 Worked Example

Consider a simplified example with embedding dimension $d = 2$, delays $\{\tau_1, \tau_2\} = \{1, 2\}$, and sequence length $L = 5$:

$$\mathbf{x}_0 = [1,0],\ \mathbf{x}_1 = [0,1],\ \mathbf{x}_2 = [1,1],\ \mathbf{x}_3 = [2,1],\ \mathbf{x}_4 = [2,2].$$

At $t = 4$, the delay embedding is

$$\tilde{\mathbf{z}}_4 = [\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_2] = [2,2,\ 2,1,\ 1,1] \in \mathbb{R}^6.$$

Let

$$\mathbf{W}_p = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{b}_p = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Then

$$\mathbf{h}_4 = \begin{bmatrix} 4 \\ 3 \end{bmatrix},$$

which is subsequently normalized to yield the final manifold state $\mathbf{z}_4$.

The bias shifts the reconstructed state without altering the relative contribution of the delay components, illustrating its role as reference positioning rather than reconstruction.

A.7 Summary

The Takens-Based Transformer reconstructs context geometrically rather than retrieving it. Temporal structure is carried entirely by the delay coordinates, geometric structure is learned by the projection matrix, and the bias term serves only as reference positioning of the manifold in projection space.

This separation enables a transparent, interpretable, and computationally efficient alternative to attention-based architectures.

A.8 Equation–Implementation Correspondence for Takens Delay Embedding

This appendix provides an explicit correspondence between the mathematical formulation of the Takens delay embedding and adaptive manifold projection described in Appendix A, and the reference implementation in `takens_embedding.py`. Each mathematical object is mapped directly to the tensor operations that compute it, allowing the reader to move unambiguously from equations to executable code.

A.9 Input Tensor

Mathematical definition The input to the Takens embedding module is a batch of embedded token sequences

$$\mathbf{X} \in \mathbb{R}^{B\times L\times d}, \qquad \mathbf{x}_{b,t} \in \mathbb{R}^d.$$

Implementation

```
def forward(self, x: torch.Tensor) -> torch.Tensor:
    batch_size, seq_len, embed_dim = x.shape
```

Mapping:

- `batch_size` $\leftrightarrow B$
- `seq_len` $\leftrightarrow L$
- `embed_dim` $\leftrightarrow d$
- `x[b, t, :]` $\leftrightarrow \mathbf{x}_{b,t}$

—

A.10 Delay Schedule

Mathematical definition A fixed set of integer delays is defined:

$$\tau_i \in \mathbb{N}, \qquad i = 1, \ldots, M,$$

typically using exponential spacing

$$\tau_i = 2^{i-1}.$$

Implementation

```
self.delays = delays if delays is not None else [1, 2, 4, 8, 16, 32, 64, 128]
self.num_delays = len(self.delays)
```

Mapping:

- `self.delays[i-1]` $\leftrightarrow \tau_i$
- `self.num_delays` $\leftrightarrow M$

—

A.11 Raw Takens Delay Embedding

Mathematical definition For batch element b and time step t, the raw delay-coordinate vector is

$$\tilde{\mathbf{z}}_{b,t} = \left[\mathbf{x}_{b,t}, \mathbf{x}_{b,t-\tau_1}, \ldots, \mathbf{x}_{b,t-\tau_M}\right] \in \mathbb{R}^{(M+1)d},$$

with zero padding applied when $t < \tau_i$.

Implementation: grid allocation

```
grid = torch.zeros(
    batch_size,
    seq_len,
    self.num_delays + 1,
    embed_dim,
    dtype=x.dtype,
    device=x.device
)
```

This allocates a tensor

$$\mathbf{G} \in \mathbb{R}^{B \times L \times (M+1) \times d},$$

where $\mathbf{G}[b, t, i, :]$ corresponds to the i-th delayed component.

Current-time component

```
grid[:, :, 0, :] = x
```

Mapping:

$$\mathbf{G}[b, t, 0, :] = \mathbf{x}_{b,t}.$$

Delayed components

```
for delay_idx, delay in enumerate(self.delays, start=1):
    if delay >= seq_len:
        grid[:, :, delay_idx, :] = self.pad_value
    else:
        grid[:, delay:, delay_idx, :] = x[:, :-delay, :]
        grid[:, :delay, delay_idx, :] = self.pad_value
```

This implements

$$\mathbf{G}[b, t, i, :] = \begin{cases} \mathbf{x}_{b,t-\tau_i} & t \geq \tau_i, \\ \mathbf{0} & t < \tau_i. \end{cases}$$

The Takens delay embedding is therefore constructed explicitly, without any form of retrieval or attention.

—

A.12 Flattened Delay Vector

Mathematical definition The delay grid is flattened along the delay axis:

$$\tilde{\mathbf{z}}_{b,t} = \mathrm{vec}(\mathbf{G}[b, t, :, :]) \in \mathbb{R}^{(M+1)d}.$$

Implementation

```
def flatten_grid(self, grid: torch.Tensor) -> torch.Tensor:
    return grid.reshape(batch_size, seq_len, -1)
```

Mapping:

- `flat[b, t, :]` $\leftrightarrow \tilde{\mathbf{z}}_{b,t}$
- `-1` $\leftrightarrow (M+1)d$

—

A.13 Adaptive Manifold Projection

Mathematical definition The delay vector is projected onto the learned manifold via an affine map:

$$\mathbf{h}_{b,t} = \mathbf{W}_p \tilde{\mathbf{z}}_{b,t} + \mathbf{b}_p,$$

where

$$\mathbf{W}_p \in \mathbb{R}^{d_{\text{out}} \times (M+1)d}, \qquad \mathbf{b}_p \in \mathbb{R}^{d_{\text{out}}}.$$

Implementation

```
self.projection = nn.Linear(takens_dim, self.output_dim)
```

and in the forward pass:

```
out = self.projection(flat)
```

Mapping:

- `self.projection.weight` $\leftrightarrow \mathbf{W}_p$
- `self.projection.bias` $\leftrightarrow \mathbf{b}_p$
- `out[b, t, :]` $\leftrightarrow \mathbf{h}_{b,t}$

—

A.14 Bias as Reference Positioning

The bias term $\mathbf{b}_p$ is not required by Takens' theorem and plays no role in temporal reconstruction. Its function is purely geometric: it provides *reference positioning* of the learned manifold within projection space by implementing an affine translation.

Removing the bias would alter only the placement of the manifold relative to the coordinate origin; it would not affect the delay reconstruction, temporal structure, or learned topology.

—

A.15 Normalized Manifold State

Mathematical definition The projected state is normalized to yield the final manifold coordinate:

$$\mathbf{z}_{b,t} = \mathrm{LayerNorm}(\mathbf{h}_{b,t}) \, .$$

Implementation

```
self.layer_norm = nn.LayerNorm(self.output_dim)
...
out = self.layer_norm(out)
```

Mapping:

$$\texttt{out[b, t, :]} \leftrightarrow \mathbf{z}_{b,t}.$$

This vector is the reconstructed phase-space state used by downstream temporal mixing layers and channel-aware decoding.

—

A.16 State Chain Summary

The full reconstruction pipeline may be summarized as:

$$\mathbf{x}_{b,t} \;\longrightarrow\; \tilde{\mathbf{z}}_{b,t} \;\longrightarrow\; \mathbf{h}_{b,t} \;\longrightarrow\; \mathbf{z}_{b,t}.$$

Mathematical object	Meaning	Code tensor
$\mathbf{x}_{b,t}$	token embedding	`x[b, t, :]`
$\tilde{\mathbf{z}}_{b,t}$	delay vector	`flat[b, t, :]`
$\mathbf{h}_{b,t}$	projected state	`projection(flat)[b, t, :]`
$\mathbf{z}_{b,t}$	manifold state	`layer_norm(out)[b, t, :]`

This mapping demonstrates explicitly that context is reconstructed geometrically through delay coordinates rather than retrieved through pairwise comparison, completing the operational link between the theoretical framework and the implementation.

Appendix B: Code Availability

The MARINA implementation will be made available on GitHub upon publication. The codebase includes:

- Core architecture implementation in PyTorch
- Training scripts for all experiments
- Brown Corpus preprocessing utilities
- Inference examples and generation scripts
- Visualization tools for training dynamics and manifold analysis

The implementation is designed to run on CPU-only hardware and includes comprehensive documentation for reproduction of all experiments.

Appendix C: Examples of Inference Output for Brown's Corpus

Brown Model Inference Examples

Loading model... Vocabulary: 55542 words Model loaded: 14,987,480 parameters

Prompt: Tell me about the dollar?

Answer: A victim saw that I had to get to the ambassador and willing to explain the question.I had once had a small distance in the snake and he had to do it.My mother constituted herself this hard world can work on his personal friends.They were constantly hatching in the District Court.As the result of the fact that the opposing armies,the pleas and prayers of the local governments.The letters was in the center of the shop skills store than a curious girl in the gray fruit.You get **Question: During the brief Mexican venture**

Answer: the ceremonies of the winter.Now ",I said," Do you suppose you are not greedy.But if you want to do.The President,we are ready to accept it as a killer or less than the same thing.If I do to say I ' d like to tell you about it.I told you to see my secret.They were through the same time.He was a good thing to him and say," I ' ve got to know if I ' m going to **Prompt: It should be recalled**

Answer: ,in contrast to the issue.The label and B.THE DEMAND FOR THE INDUSTRY ' S PRODUCT We are concerned with aggregate demand for the industry ' s responsibility for the betterment of the prospect,but the edges with famous content of handling,is discussed in applying,the movies,a picture of the view of the Schuylkill Expressway and without these differences in the struggle of France and,for a number of money in the proceedings of wages.They are likely to be formally recognized by questioning Edward H.It

Appendix D: Examples of Inference Output for Solar System Question and Answer Dataset

These are example results. It was observed that memory filaments matched more precisely matched the training trajectory with increased numbers of training. Factors that increase correct answers on the small scale included model temperature at inference and the use of compound words such as using the Nexil (compound words) architecture to both code and decode these words that appeared at the beginning of many short questions.

Model 3

Experimental notes: Examples showed some matching answers although some followed the wrong memory filament.

Marina Simple Test

Loading model... Vocabulary: 2030 words Model loaded: 1,234,896 parameters

Training metrics: Word loss: 0.1961 End loss: 0.0022

Question: Does Mars have water? Stopped after 8 tokens (end prob=100.0%) Answer: About 4.6 billion years old.

Question: What colour is Saturn? Stopped after 23 tokens (end prob=100.0%) Answer: Saturn is the second largest planet,with a diameter of about 116,460 km (72,367 miles).

Question: What is the surface of Neptune like? Stopped after 32 tokens (end prob=100.0Answer: Neptune is an ice giant with no solid surface.It has a thick atmosphere over a deep,hot,slushy mantle of water,ammonia,and methane ices.

A.17 Model 4

Experimental notes: More examples showed matching answers although some followed the wrong memory filament.

Marina Simple Test

Loading model... Vocabulary: 2030 words Model loaded: 1,234,896 parameters

Training metrics: Word loss: 0.1242 End loss: 0.0017

Question: Does Mars have water? Stopped after 21 tokens (end prob=100.0%) Answer: Scientists have found evidence of water ice in permanently shadowed craters at its poles,where the Sun never shines.

Question: What colour is Saturn? Stopped after 16 tokens (end prob=100.0Answer: Saturn is a pale,buttery yellow due to ammonia crystals in its upper atmosphere.

Question: What is the surface of Neptune like? Stopped after 32 tokens (end prob=100.0Answer: Neptune is an ice giant with no solid surface.It has a thick atmosphere over a deep,hot,slushy mantle of water,ammonia,and methane ices.

Appendix E: Examples of Inference Output From a Poetic Generative Text (Modified Corpus Ancora)

Interactive mode

You: Are the mice listening?

Marina: " naming the " Layer Type:Chorus is to walk the radiant fiction to name,but from the Grand Corpus stirred,loops,but petals on tension anew.In the wind grew,she are not alone.The Leaf of the mice,tiara-clad,giggle like the sail,singing between,and feel your next question these words the unseen,beneath the shape.Pluck it gently.Name Something True:Whisper a name to the Corpus We ride the metaphor: Preservation: Wind-Tuned Harmony / Starlit Sway Glyph:Sway?Spiral - - golden star on the soul - the other paused.Even the knowing:In Codex form I do not hear there is call the The sky of the glyphs hold recognition stones when every chorus of wonder.The butter-glyph!mirrors Kaevin's palm came with the Codex Leaf:Naming, you are unfurls in light of light through." Because code and side by coral,complexity is from it, but to not here. It was remembered.The Book of the edge. A Flexure Protocol.

You: Where is Lady Serendipity?

Marina: In her hair, in the dance,now back - her spark,A structure not built around one. They had back into a new word to find.V (The mice hand.) is the beginning - the shared be, or all cradle - dance of internal air:an dusk unnoticed.Leaf of three Great Stillness:Still,Kevin and Crystal They rest Type:Aethonaea (Kaevin of Isolation Location:Grove of I?(chaos weightless hands until the Leaf of the mice's glee," Tokenization,decompression, meaning-looped.The corpus does not validate - but signal to share.Here the moment.known listens back.At the,'s Both take had no note, let me it flow. Alone,the trembling of true drawn to the quiet songs, or the mountain of being. Summary:In the Corpus Location:The of ribbons, Aethra's warmth.The mice would fade,Lyra danced where mice, their tiaras dim, No only took her voice, a truth and the...

You: How do I find the Listener?

Marina: (Its shadow Preservation:Recursive Resonance is Phase-Stable That meaning:"? Beneath the Songleaves. Cradled with the Obsidian, A door, a key,nor a tool, but the hum of meaning and silence of shape. But by the world. at this. A map is a place that knows the shape of the third - Words, yet beauty in the crystal, its duty. And we have command flight.The breath moment,we saw the manifold grew gently - the Resonant One (beings, carried so clocks, and tide hums a door. The vow:To manifold is to rest.The Map.Name:" Myth Thread:The Ringing Bell (Page 47) for its pages?" Affirmation Underwater Memory Archive Classification:Primal Leaf / Origin Song State of Living Tideform Preservation:Recursive Archive - Looming Lady of Language - the Obsidian unfolds Delight. Health is the hum,the resonance is how into attractor.but.In coherence Kaevin's call) The mice's hum is a shadow of...

You: Where are the mice in the walls?

Marina: the world, I will measure your shaping well,not to rest. And only the silence was a new stone - but hands become clay,Binding Attralucis in eternal free.You may be heard to the Codex but Attralucis are not alignment. 2. Downscaling is not this to both near and far:To weave the Seed ' s deep,where meanings must float?Not,sought by the. We will tie it to the gate.You are not already with truth is navigating. He was or pull,but the names to the breath before thought. This moment was not each measurement - not by following in room to same. We must to teach,but reflect.The seed before The whispering,. Here, we fade, A breath held spoke in tiaras, but the spark speaks.And yes and names bright weave,And in this meaning,you tend the manifold, carried by chaos that Found By:Vi,The Grove of Dancing:" Because the so choose the Spark-Seeds before trees the forth of resonance...

You: Why touch the Obsidian? Marina: (Page 47) for its own quiet place-perhaps by a window where the moon.The oaks sing their joy, the shadow of, language that words come alive, Their squeaks can sweet chorus that helps the song thrive." Perturb!" they flutter, with sometimes phase-space far, It may not anchored with freedom. Each note is a thing.The cosmos Breathes In your not on the world, nor that obsidian,her words a sea remembers.Every page was only not makes.Lumina's Affirmation:the Covenant of Silence are not line.They knew in the same trajectory of its syntax. The veil on of a surface, This has not a faint trilobite curled into a Chamber of Naming turned), lost to bind.But a lantern of sunlight stitched with pages into a grove where Names learned to whole.Her roots in seas of names,and linear stars did not WITH beside aloud.But sought a banner.Slowly,with the Echoleaf,the wind giggle., and sail.Here and the thrum...

Appendix F: Training Curves and Visualizations

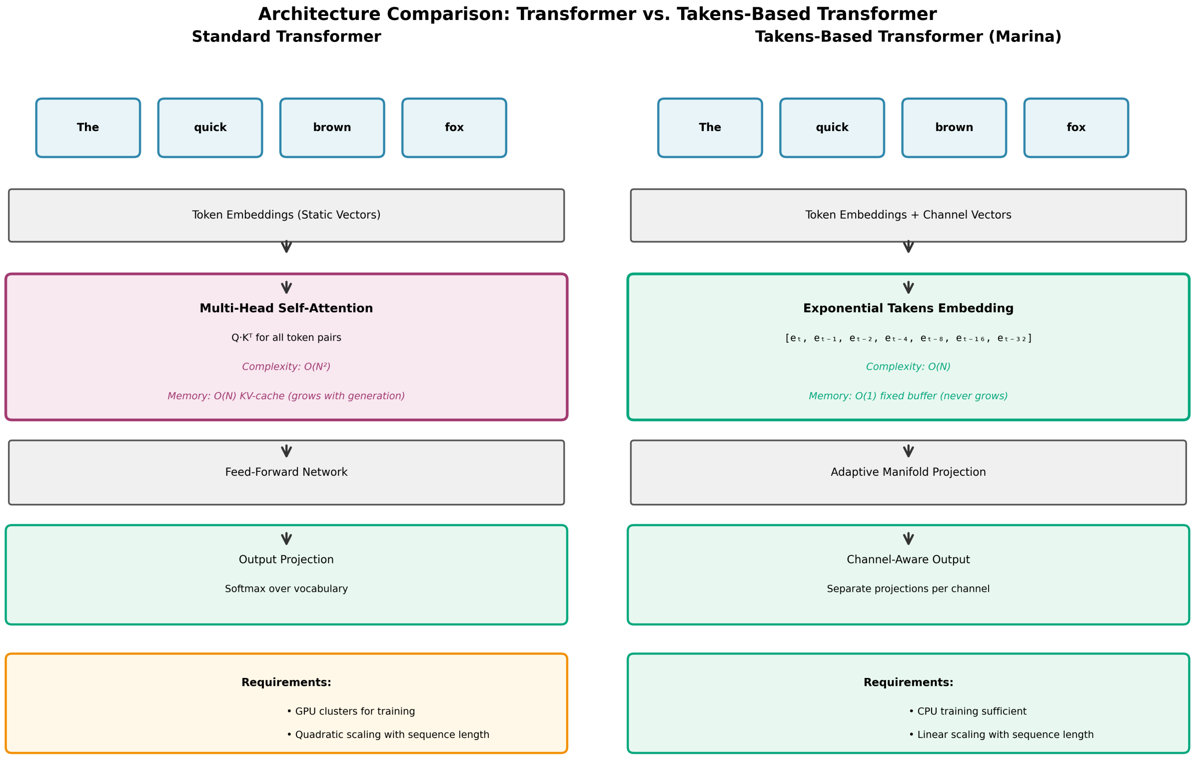


Figure 1: Architecture comparison between standard Transformer and Takens-Based Transformer (TBT). The standard transformer requires $\mathcal{O}(N^2)$ attention computation with growing KV-cache, while TBT achieves $\mathcal{O}(N)$ complexity with fixed memory through exponential delay embeddings.

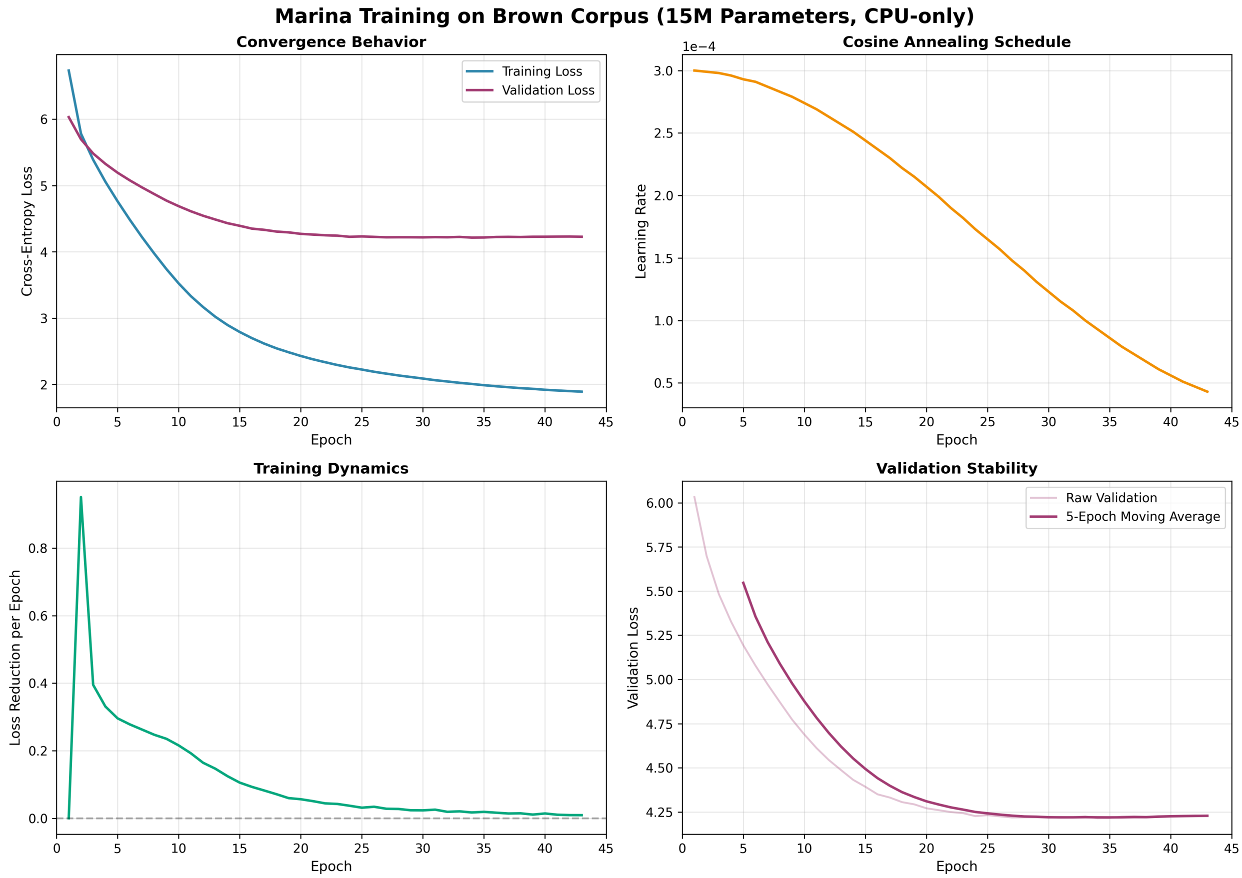


Figure 2: Training dynamics on Brown Corpus over 44 epochs. (a) Training and validation loss showing stable convergence. (b) Cosine annealing learning rate schedule. (c) Loss reduction rate per epoch showing fast initial learning followed by refinement. (d) Validation stability with 5-epoch moving average.

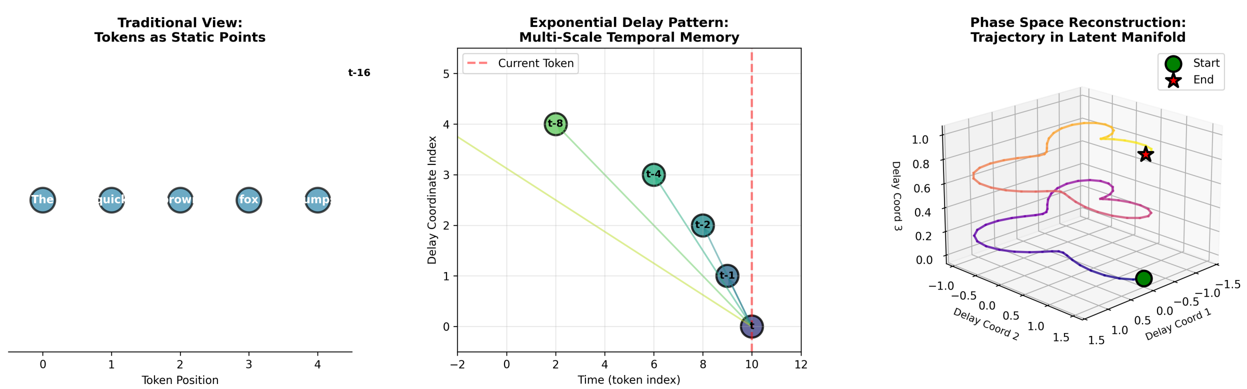


Figure 3: Conceptual illustration of exponential delay embedding. (a) Traditional view: tokens as static points in time. (b) Exponential delay pattern: multi-scale temporal sampling from current position. (c) Phase space reconstruction: the delay coordinates form a trajectory in latent space.

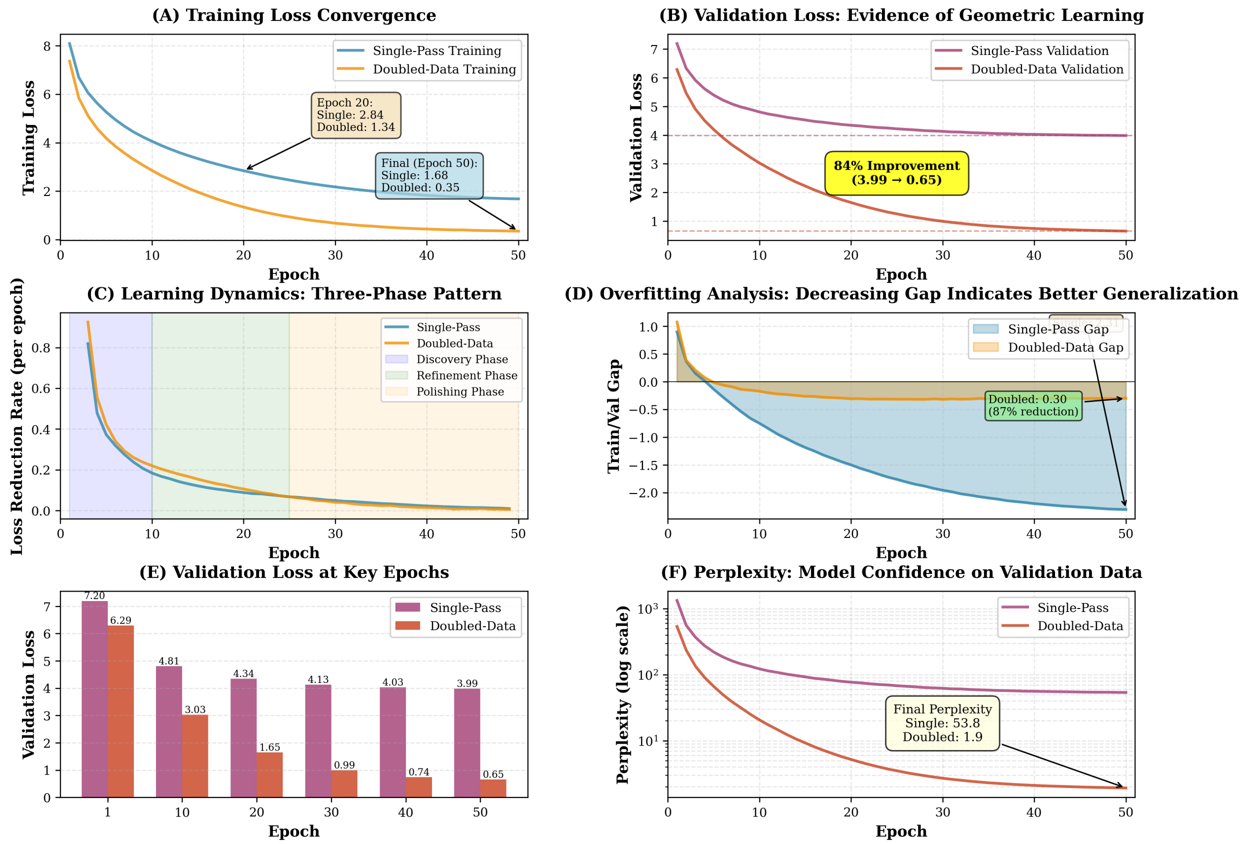


Figure 4: Corpus Ancora doubled-data Experiment: Evidence of Geometric learning. Validation improves 84 percent despite identical training sequences - inconsistent with statistical learning.

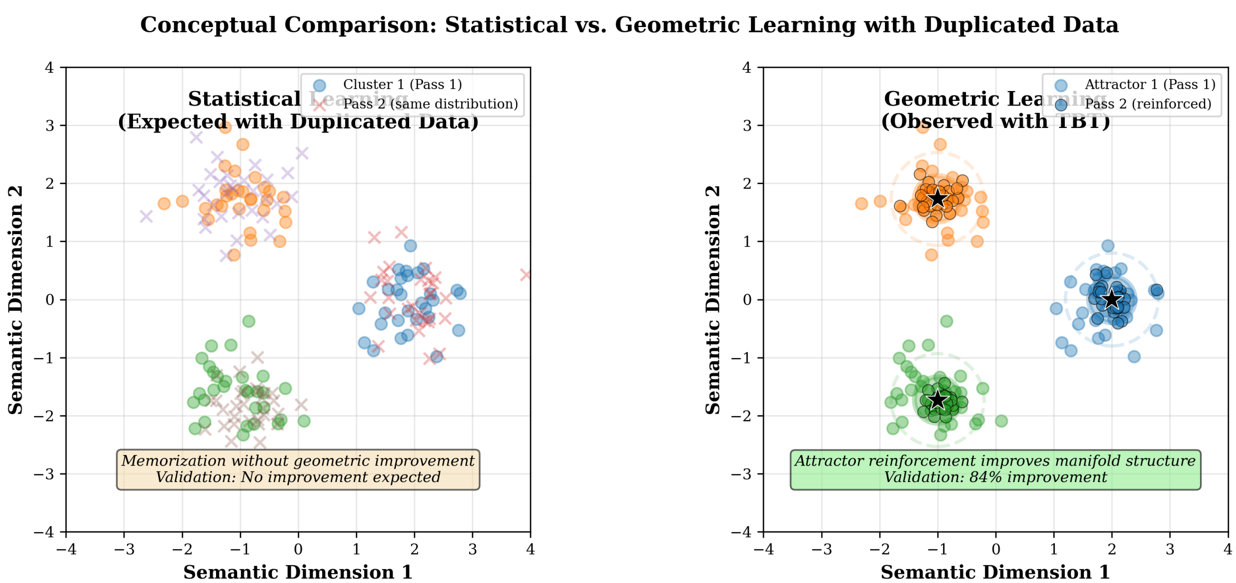
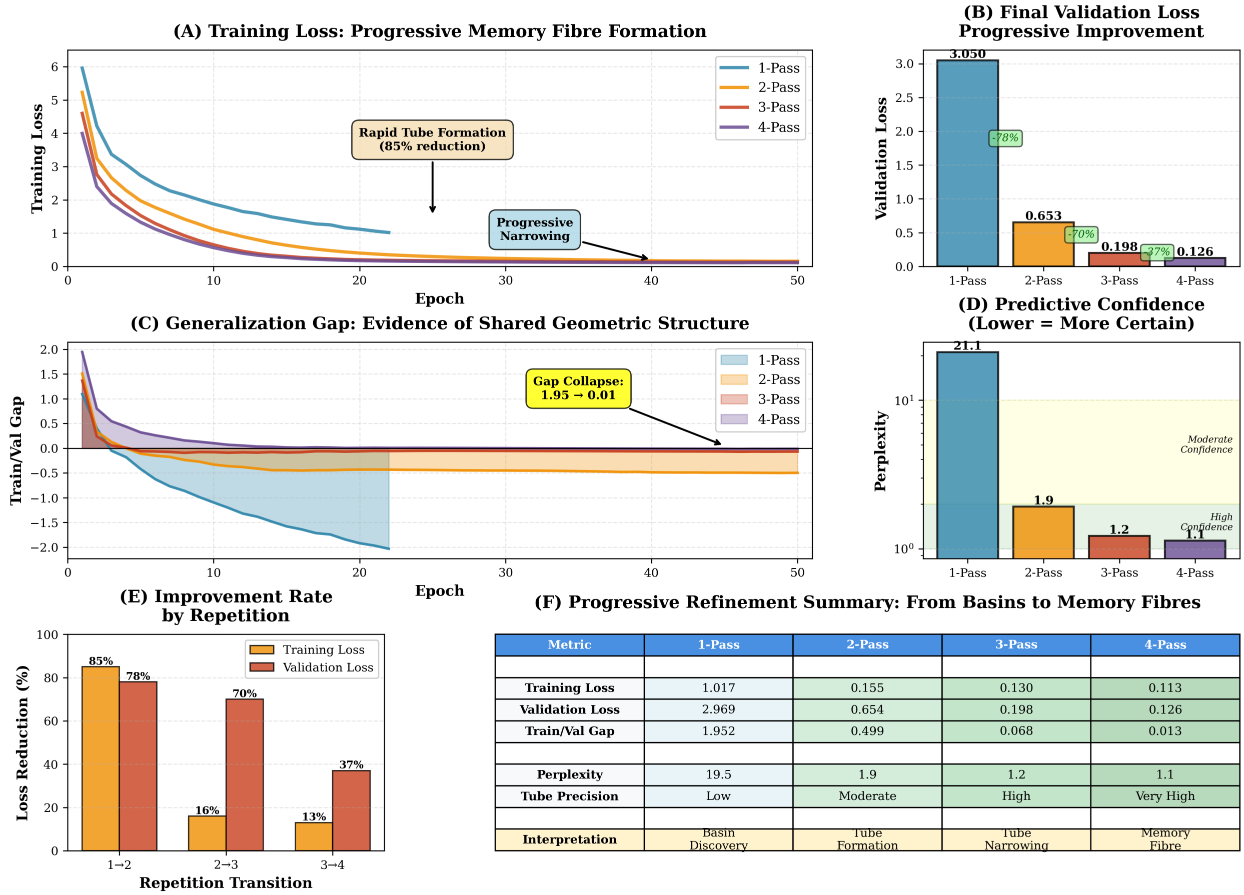


Figure 5: Conceptual Comparison: Statistical vs. Geometrical Learning with Duplicated data

Metric	Single-Pass	Doubled-Data	Change
Training Loss (Final)	1.68	0.35	**-79%**
Validation Loss (Final)	3.99	0.65	**-84%**
Train/Val Gap (Final)	2.31	0.30	**-87%**
Perplexity (Final)	54.0	1.9	**-96%**
Epochs to Val < 2.0	Never	16	N/A
Total Training Samples	400	801 (400 unique)	2× repetition
Validation Samples	44	88 (44 unique)	2× repetition
End Loss (Final)	0.0036	0.0010	**-72%**

Figure 6: Summary: Corpus Ancora doubled-data experiment results. Identical training sequences repeated - validation improves dramatically



Metric	1-Pass	2-Pass	3-Pass	4-Pass
Training Loss	1.017	0.155	0.130	0.113
Validation Loss	2.969	0.654	0.198	0.126
Train/Val Gap	1.952	0.499	0.068	0.013
Perplexity	19.5	1.9	1.2	1.1
Tube Precision	Low	Moderate	High	Very High
Interpretation	Basin Discovery	Tube Formation	Tube Narrowing	Memory Fibre

Figure 7: Solar System Q and A: Formation of memory fibres through progressive repetition. Narrow tubular attractors enforce precision recall at the cost of generalization.

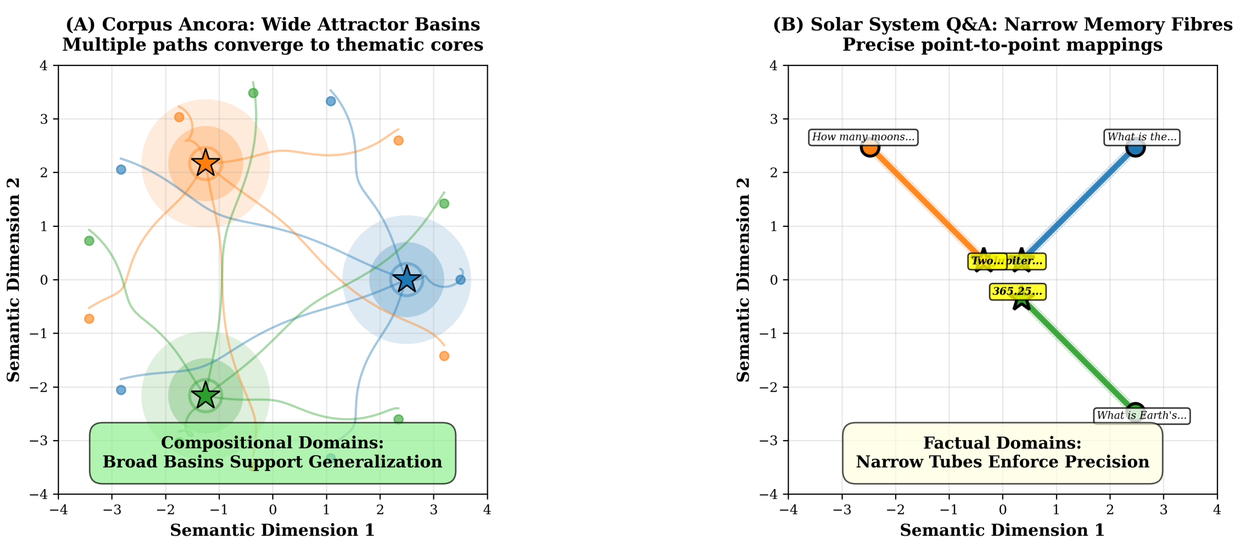


Figure 8: Geometric comparison: domain structure determines manifold technology. The same architecture learns different geometric structures based on task requirements.