From Formula to Process: Bridging Machine Learning Mathematics and Nonlinear Dynamics

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

Abstract

Modern machine learning is commonly introduced through a small set of powerful mathematical forms: affine transformations, activation functions, loss functions, gradient descent, probability distributions, and scaled dot-product attention. These forms are useful and successful descriptions. They allow engineers and researchers to reason about models at a high level of abstraction.

These formulas are not wrong. They are highly effective symbolic compressions.

However, the compression can hide something essential: no physical machine performs a formula as a whole. A processor does not instantiate an affine transformation or an attention head as a single mathematical object. It performs a finite sequence of operations—fetching, multiplying, accumulating, rounding, storing, and updating—coordinated across hardware. Even parallel execution on GPUs or TPUs is itself a carefully orchestrated set of finite state changes.

The purpose of this essay is to build a bridge between the conventional symbolic description of machine learning and a process-based, finite, measurable description grounded in Finite Mechanics. The bridge does not discard the mathematics. It unfolds it.

The central claim is simple: Modern AI is commonly described through symbolic mathematical compressions, but it is physically instantiated as finite sequential measurement and update processes.

This distinction matters because it opens a direct path from machine learning into nonlinear dynamics, time-series reasoning, and a deeper account of how meaning, learning, and attention emerge from ordered processes rather than static formulas. Companion works have already applied this lens first to the reinterpretation of attention itself (Haylett 2025a) and then to a fully operational architecture—the Takens-Based Transformer—that replaces attention with explicit delay-coordinate reconstruction (Haylett 2025b). Together they extend the Geofinite reframing and suggest that the same dynamical principles may scale to generic nonlinear dynamical systems modeling..

1 The Usefulness and Limitation of the Conventional Presentation

Standard pedagogical accounts of modern AI often begin by presenting the field as built from a small coherent set of mathematical ideas: representation, prediction, loss, probability, optimization, and attention. This is a good compression. It helps readers see that many apparently different systems share the same general pattern: represent data numerically, transform it through parameterized functions, compare the output to a target, and update the parameters to reduce error.

This algebraic surface is valuable. But once the equations are introduced, the reader may begin to treat the equation *as* the process. The finite, time-ordered operations that actually execute on hardware fade into the background.

A neural layer is commonly written as

$$z = Wx + b.$$

This is mathematically clean. It tells us that an input vector x is transformed by a weight matrix W, shifted by a bias vector b, and produces an output vector z. At the symbolic level, this is elegant and compact.

But at the machine level, there is no single act called "performing $Wx + b$." There is only a finite sequence of operations. For each output coordinate the processor must perform something like

$$z_i = \sum_j W_{ij} x_j + b_i.$$

This expands into a sequence of multiplications, accumulations, finite-precision rounding, memory movement, and storage. The compact symbolic expression $z = Wx + b$ is therefore a compression of many ordered finite operations.

In Geofinite terms, the formula is a symbolic compression of a measured process.

2 The Formula Is Not the Physical Operation

This distinction is not merely philosophical. It matters for how we understand learning.

In classical mathematical language, a function is often treated as if it maps an input to an output as a completed object: $f(x) = y$. But in a physical machine, $f(x)$ is not simply "there." It must be unfolded. It must be generated.

The function is enacted as a process. The input must be encoded. The parameters must be accessed. The operations must occur in some order. Intermediate values must be held. Numerical approximations must be made. The output must be constructed.

This means that every machine learning function has two descriptions:

First, the symbolic compression: $f_\theta(x)$.

Second, the operational unfolding: $x \to$ encode $\to$ multiply $\to$ accumulate $\to$ activate $\to$ store $\to$ propagate.

The first is useful for mathematical reasoning. The second is closer to what the machine actually does.

A mature account of AI needs both. The symbolic form is the map. The process is the measured terrain.

3 The Loss Landscape Is Not Simply Given

Machine learning is often described as search across a loss landscape. The loss function is the terrain and gradient descent is the way the model navigates it.

This is a useful image. But from a process-based perspective, it is incomplete.

The machine never sees the whole landscape. The full loss landscape of a modern model is too large to inspect. It samples it.

Each forward pass measures a relation between an input, the current parameter state, and an output. Each loss calculation measures a discrepancy between prediction and target. Each gradient calculation measures a local direction of change. Each optimizer step modifies the very structure that will be used in the next measurement.

Thus training is not simply navigation through a known terrain. It is an iterative construction of local terrain measurements.

The landscape is revealed through sequential measurement.

In the conventional view:

$$\theta^* = \arg\min_\theta L(f_\theta(x), y).$$

In the process view:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L_t.$$

But even this remains compressed. More explicitly:

$$\theta_t \to f_{\theta_t}(x_t) \to \hat{y}_t \to L(\hat{y}_t, y_t) \to \nabla_\theta L_t \to \Delta\theta_t \to \theta_{t+1}.$$

Here, training is a time-indexed sequence of measurements and updates. The subscript t matters. Without t, the equation looks like timeless optimization. With t, it becomes a dynamical process.

4 Learning as Sequential Measurement

A model does not learn because a formula exists. It learns because a repeated finite process occurs.

At each step, the system performs an operation that can be described as:

$$\text{Measure} \rightarrow \text{Compare} \rightarrow \text{Update}.$$

This can be written as a process loop:

$$M_t = \text{measurement of model behaviour at step } t, \quad E_t = \text{measured discrepancy or error}, \quad U_t = \text{update derived from that discrepancy}$$

The model is therefore not simply a parameterized function. It is a finite symbolic measuring structure undergoing iterative transformation.

This reframes learning. The model is not merely "fitting data." It is reshaping its own future measurement responses by repeatedly encountering finite symbolic differences.

This gives us a bridge to nonlinear dynamics. A learning system is a state-evolving system. Its next state depends on its current state, its input, its measured error, and its update rule:

$$S_{t+1} = F(S_t, x_t, y_t, E_t)$$

where S_t includes parameters, optimizer state, activation patterns, memory structures, and other relevant machine states.

From this perspective, machine learning is not only algebraic optimization. It is a nonlinear time-evolving process.

5 Why "Affine Transformation" Can Disguise the Machinery

The phrase "affine transformation" is mathematically correct. But it can disguise the machinery by compressing a process into an object-like phrase.

An affine transformation sounds like a geometric act: a vector is rotated, stretched, shifted, or projected. This is useful at the level of abstract geometry. But in a physical AI system, the transformation is generated through repeated finite operations.

The symbolic compression

$$z = Wx + b$$

hides

$$z_i = (((W_{i1}x_1 + W_{i2}x_2) + W_{i3}x_3) + \dots) + b_i.$$

And each of these steps hides further processes: representation, rounding, finite precision, memory movement, clock cycles, hardware scheduling, and energy expenditure.

The formula is therefore not the physical event. It is a compact symbolic trace of the event.

This has a deeper implication. If the symbolic language hides the process, then it can also hide where meaning, instability, emergence, and failure modes enter the system. Many important phenomena in AI may not be visible if we only look at the compressed mathematical object.

The process matters.

6 From Static Functions to Time Series

The key bridge to nonlinear dynamics is the recognition that machine learning systems unfold as time series.

A forward pass is a sequence. A training loop is a sequence. A generated text is a sequence. Attention operates over sequences. Optimizer states evolve through sequences. Model behaviour across prompts forms a sequence. Even the user-model conversation is a coupled symbolic time series.

A nonlinear dynamical system is often described in the form

$$s_{t+1} = F(s_t)$$

or, with input,

$$s_{t+1} = F(s_t, x_t).$$

This is directly relevant to machine learning. A model during inference may be described as

$$h_{t+1} = F_\theta(h_t, x_t)$$

where h_t is the hidden state or evolving internal representation, x_t is the current input token or symbol, and F_θ is the learned transformation.

For transformer-based models, the state is not a simple recurrent hidden vector in the traditional RNN sense. But the generated sequence still has temporal dependence. The next token distribution depends on the preceding symbolic context:

$$p(x_{t+1} \mid x_1, x_2, \ldots, x_t).$$

This is a time-series process.

The model does not generate a paragraph as a static object. It generates one token after another. Each token changes the context. Each context changes the next probability distribution. The output is a trajectory through symbolic state space.

Thus, language generation is not merely classification over vocabulary. It is finite symbolic trajectory formation.

7 Attention as Relational Measurement

The conventional attention formula is

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

This is mathematically correct and computationally convenient. But it is a compression.

Operationally, the mechanism performs a set of finite relational measurements. Each query vector q_i is compared with every key vector k_j via the dot product $q_i \cdot k_j$. The resulting scores are scaled and normalized, then used to form a weighted sum of value vectors. In expanded process language this is a sequence:

$$s_{ij} = q_i \cdot k_j, \quad \tilde{s}_{ij} = \frac{s_{ij}}{\sqrt{d_k}}, \quad a_{ij} = \frac{\exp(\tilde{s}_{ij})}{\sum_l \exp(\tilde{s}_{il})}, \quad o_i = \sum_j a_{ij} v_j.$$

Each dot product is a measured geometric alignment between two symbolic states at different positions in the sequence. The entire block is therefore a pairwise comparison across a finite context window.

This operation is structurally identical to the method of delays used in nonlinear dynamical systems to reconstruct a latent attractor from a single observable time series (Takens 1981; Packard et al. 1980). In delay embedding, a scalar sequence $x(t)$ is mapped into a higher-dimensional space by forming vectors of the form

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

where m is the embedding dimension and τ is the delay. Takens' theorem guarantees that, for sufficient m, the reconstructed trajectory is diffeomorphic to the original attractor—preserving its geometric structure.

In a transformer, the learned projections W_Q and W_K play the role of the delay coordinates, while the context window supplies the finite history. The similarity matrix A_{ij} is therefore a discrete, high-dimensional analogue of a phase-space trajectory. The model does not "attend" in any cognitive sense; it reconstructs a surrogate manifold in which sequential symbolic data becomes geometric relationships. (For a full formal equivalence and the resulting architectural simplifications—removal of explicit positional encodings and softmax as redundant once the attractor geometry is explicit—see Haylett 2025a.)

The conventional account says: attention finds relevance. The dynamical account says: attention reconstructs relational geometry from a finite symbolic time series.

The natural next step is to make the reconstruction explicit. Once attention is understood as an implicit form of delay-coordinate embedding, the Geofinite perspective immediately suggests replacing the quadratic relational search with a direct, finite, sequential process of phase-space reconstruction. This is precisely what the Takens-Based Transformer (TBT) does. Introduced in Haylett (2025b) and implemented as the MARINA architecture (Manifold-Aware Reconstruction and Inference Network Architecture), the TBT treats language (or any time-series) as a trajectory on a learned semantic manifold. Context is no longer retrieved; it is reconstructed at every step through exponentially spaced delay coordinates:

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

These delays are fed into a learned adaptive projection layer that maps the high-dimensional delay vector onto a dense manifold state $\mathbf{z}_t$. No attention matrix, no growing KV-cache, no quadratic cost. Memory is $\mathcal{O}(1)$ and per-token complexity is $\mathcal{O}(\log N)$. The model evolves the trajectory forward by steering $\mathbf{z}_t$ through learned temporal mixing layers, exactly as a nonlinear dynamical system evolves according to its vector field.

The same architecture is immediately generalizable beyond language. Any observable time-series from a physical, biological, or engineered nonlinear dynamical system can be treated as the "token" stream. The delay-coordinate reconstruction recovers the latent attractor geometry, and the manifold projection learns the flow rules. In this sense the TBT is not merely a language model; it is a generic finite symbolic dynamical system simulator. Training on sensor readings, financial ticks, EEG signals, or climate data becomes a matter of reinterpreting the input vocabulary and channel topology—no architectural changes required.

This is the direct payoff of the process view developed throughout the essay. Every formula we unfolded—affine layers, loss, gradients, attention—has now been reimplemented as an explicit, measurable, time-indexed sequence of state updates on a physical substrate. The symbolic compression remains useful for high-level reasoning; the unfolded process becomes the operational engine.

8 The Missing Time Index in Machine Learning Language

Many machine learning formulas suppress time.

This is understandable because the mathematics aims for compactness. But suppressing time can hide the sequential nature of the process.

For example:

$$z = Wx + b$$

is really

$$z_t = W_t x_t + b_t$$

during training, because W and b change over update steps.

Similarly:

$$L(f_\theta(x), y)$$

is really

$$L_t(f_{\theta_t}(x_t), y_t)$$

and gradient descent

$$\theta \leftarrow \theta - \eta \nabla_\theta L$$

is really

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta L_t.$$

The t index restores the process.

It reminds us that learning is not a static relation between variables. It is an ordered sequence of finite measurements and state changes.

This is where nonlinear dynamics enters naturally. Once the time index is restored, the system is no longer just a collection of formulas. It becomes a trajectory.

9 The Machine as a Finite Measuring Apparatus

From this perspective, the machine learning model can be understood as a finite measuring apparatus.

It measures input relations. It measures prediction error. It measures local gradients. It measures token compatibility through attention. It measures distributional mismatch through cross-entropy or KL divergence. It measures uncertainty through entropy. It measures alignment through dot products or cosine similarity.

But each of these "measurements" is not a direct contact with reality. It is a finite symbolic operation inside a constructed system.

An embedding is not meaning itself. It is a finite symbolic measurement-like projection produced by a trained system. Cosine similarity is not the whole geometry of meaning. It is one observable taken from that geometry.

This matters because it prevents overclaiming. We do not need to say that the model "understands" in a vague or mystical sense. Nor do we need to reduce it to mere mechanical lookup. We can say something more precise:

A model constructs finite symbolic measurements over learned relational structures, and its outputs are trajectories through those structures.

That is a bridge both engineers and philosophers can cross.

10 Nonlinear Dynamics: The Natural Language of Process

Once machine learning is viewed as sequential measurement and update, nonlinear dynamics becomes a natural explanatory framework.

A nonlinear dynamical system evolves through time according to rules in which outputs are not simply proportional to inputs. Small changes in initial state can lead to different trajectories. Systems may settle into attractors, move between basins, exhibit transient behaviour, or become unstable.

This language maps well onto machine learning behaviour.

A prompt can be understood as an initial perturbation. The model's response is a trajectory. The context window is a finite container. Attention constructs relational measurements inside that container. Sampling introduces controlled variation. Temperature changes the sharpness of transition probabilities. Repetition, collapse, hallucination, and sudden shifts in tone can be viewed as trajectory phenomena rather than isolated output errors.

In this view, a language model is not best understood as a static database of probabilities. It is a finite symbolic dynamical system whose responses unfold through constrained trajectories.

This does not deny the statistical nature of the model. It situates statistics inside process.

11 A Bridge Statement for Machine Learning Researchers

For someone working in ML, the bridge can be stated gently:

The standard mathematical account of machine learning is correct as a compressed engineering description. However, each formula denotes a finite operational process implemented through sequential state changes. When these processes are indexed over time, machine learning systems can be understood as nonlinear dynamical systems that construct, measure, and update symbolic trajectories.

Or more compactly:

Machine learning formulas describe what is compressed; process dynamics describe how it is physically unfolded.

This bridge preserves the ML account while making room for deeper analysis.

It does not say: "Your equations are wrong."

It says: "Your equations are compressed process descriptions. Let us now unfold them."

12 Worked Example: Unfolding an Affine Layer

Compressed form:

$$z = Wx + b.$$

Expanded coordinate form:

$$z_i = \sum_j W_{ij} x_j + b_i.$$

Sequential process form:

$$a_{i1} = W_{i1} x_1, \quad a_{i2} = a_{i1} + W_{i2} x_2, \quad \ldots, \quad z_i = a_{in} + b_i.$$

Machine-process interpretation: Fetch W_{ij}. Fetch x_j. Multiply. Accumulate. Repeat. Add bias. Store result. Pass result to next operation.

Geofinite interpretation: The affine transformation is a symbolic compression of a finite sequential measurement-and-accumulation process.

13 Worked Example: Unfolding Attention

Compressed form:

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

Expanded process: For each query q_i, compare it with every key k_j:

$$s_{ij} = q_i \cdot k_j, \quad \tilde{s}_{ij} = \frac{s_{ij}}{\sqrt{d_k}}, \quad a_{ij} = \frac{\exp(\tilde{s}_{ij})}{\sum_l \exp(\tilde{s}_{il})}, \quad o_i = \sum_j a_{ij} v_j.$$

Process interpretation: Attention is not a single act of relevance. It is a finite sequence of relational measurements, normalization, and recombination.

Dynamical interpretation: Attention reconstructs local symbolic geometry from the finite history available in the context.

14 Worked Example: Unfolding Training

Compressed form:

$$\theta^* = \arg\min_\theta L(f_\theta(x), y).$$

Sequential training form:

$$\theta_0 \to \theta_1 \to \theta_2 \to \cdots \to \theta_T.$$

Each step:

$$\hat{y}_t = f_{\theta_t}(x_t), \quad E_t = L(\hat{y}_t, y_t), \quad g_t = \nabla_\theta E_t, \quad \theta_{t+1} = \theta_t - \eta g_t.$$

Process interpretation: Training is finite sequential measurement of discrepancy followed by constrained structural update.

Dynamical interpretation: Training is a trajectory through parameter-state space, driven by measured error signals.

15 The Larger Reframing

The conventional ML statement might be:

Modern AI is function composition, optimized by gradients, under a probabilistic objective.

A process-based Geofinite statement might be:

Modern AI is finite sequential symbolic measurement, accumulated through machine processes, compressed into mathematical functions, and updated through time-indexed error-driven trajectories.

Or, with nonlinear dynamics made explicit:

Modern AI is a finite symbolic dynamical system in which learned transformations, relational measurements, and probabilistic constraints generate trajectories through a high-dimensional container space.

This is not a rejection of the standard account. It is a deeper operational layer beneath it.

16 Why This Bridge Matters

This bridge matters because many current explanations of AI stop at the formula. They show the symbolic compression but not the unfolding.

That can lead to several misunderstandings.

First, it can make AI seem more abstract than it is. The model appears to operate in a realm of pure mathematics, rather than through finite physical processes.

Second, it can hide the role of measurement. Loss, gradient, similarity, attention, entropy, and probability are all treated as mathematical objects, but each is also an operational measurement inside the machine.

Third, it can obscure time. Training, inference, generation, and interaction all unfold sequentially, yet the formulas often suppress the time index.

Fourth, it can make nonlinear behaviour seem surprising. But if we understand models as time-evolving systems, then attractors, instabilities, basins, transitions, collapse, and emergence become expected phenomena.

Finally, it can limit safety analysis. If a model is treated only as a static function approximator, then failures are often framed as bad outputs. If it is treated as a dynamical system, failures can be studied as trajectory failures: basin capture, instability, over-steering, attractor collapse, or pathological recurrence.

17 The Takens-Based Transformer: A Concrete Dynamical Realization

The bridge from conventional machine learning to nonlinear dynamics is now complete. The TBT operationalizes every insight of the preceding sections:

- Formulas are unfolded into finite sequential operations → explicit delay embedding replaces the compressed attention formula.
- Learning is sequential measurement and update → each forward pass reconstructs the current manifold state from measured history.
- The loss landscape is sampled, not known → the manifold projection learns local curvature from observed trajectories.
- Time indices matter → every step advances the trajectory by one discrete update.
- Meaning emerges from ordered processes → semantics are geometric flows on the learned attractor.

Empirical results on modest hardware (15 M parameters, CPU-only) confirm viability: stable convergence on the Brown Corpus (training loss 1.88, validation 4.21), precise factual recall via "memory fibres" (narrow tubular attractors) in question-answering tasks, and stylistically coherent long-form generation in mythopoetic domains. Channel Theory is proposed as a way to introduce structural separation between user input, internal processing, and system output, potentially offering forms of control that cannot be achieved through prompting alone..

For generic nonlinear dynamical systems modeling, the architecture is immediately applicable. Replace the token vocabulary with quantized sensor readings or symbolic observables; the same delay-reconstruction + manifold-projection loop recovers the underlying attractor and predicts future states.The TBT may therefore be viewed as a candidate finite symbolic simulator for observed nonlinear systems whose dynamics can be represented as time series.".

18 Conclusion

The mathematics of modern AI is powerful because it compresses vast operational complexity into elegant symbolic forms. But those forms should not be mistaken for the physical process itself.

An affine transformation is not performed as a single object. It is unfolded through finite operations. A loss landscape is not fully known. It is sampled through measurement. Gradient descent is not abstract movement through a pre-existing terrain. It is a sequential update process driven by locally measured discrepancies. Attention is not merely a formula for relevance. It is a relational measurement process over a finite symbolic sequence—now realized explicitly in the Takens-Based Transformer.

The formulas are useful. But the process is primary.

For machine learning researchers, this offers a constructive extension of the standard view. The equations remain valid, but they are reinterpreted as symbolic compressions of finite sequential machinery. The Takens-Based Transformer shows what happens when we take that reinterpretation seriously: we obtain leaner, more interpretable, geometrically grounded models that scale the same principles to any nonlinear dynamical system.

For nonlinear dynamics, this opens a bridge into AI: models can be studied not only as functions, but as evolving systems that generate trajectories through symbolic state space.

For Geofinitism and Finite Symbolic Mechanics, this bridge is essential. It shows that AI mathematics is not floating above the machine. It is grounded in finite measurement, finite representation, finite sequence, and finite update.

The formula is the compressed mark. The machine is the unfolding process. The landscape is not simply found. It is measured into being—through explicit phase-space reconstruction.

References

[1] Haylett, K. R. (2025a). Pairwise Phase Space Embedding in Transformer Architectures. Preprint, available at `https://finitemechanics.com/papers/phase-space-transformers.pdf`.

[2] Haylett, K. R. (2025b). Introducing the Takens-Based Transformer. Preprint (December 2025, v1.1), available at `https://finitemechanics.com/papers/takens-transformer.pdf`.

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