The Takens-Based Transformer as a General Trajectory Classification Architecture

Single-Channel and Multi-Channel Classification through Delay-Embedded Manifolds

A research monograph within the Functional Symbolic Mechanics and Geofinitist programmes

Kevin R. Haylett

Manchester, UK
July 2026

Abstract

This monograph develops the proposition that the Takens-Based Transformer (TBT) may be used not only as a language-generation architecture, but as a general system for classifying measurable trajectories. The central idea is that a time series can be unfolded through delay embedding, represented as a trajectory in a reconstructed manifold, and associated with one or more labelled channels. A channel may correspond to a single class, operational regime, functional condition, measurement context, or other admissible distinction. Classification then becomes the registration of a trajectory against learned channel basins rather than merely the calculation of a conventional class score.

Two principal architectures are developed. In the first, a single labelled channel identifies one classification dimension, such as rhythm class, machine condition, speaker identity, or movement type. In the second, multiple channels represent independent or partially dependent classification dimensions, such as signal quality, source identity, operating state, anomaly status, and final class. Multi-channel classification may be arranged in parallel, hierarchically, sequentially, or as a tree of manifold filters.

The monograph gives a formal description of training and inference, considers labelled support vectors, class prototypes, basin geometry, trajectory energy, uncertainty, out-of-distribution behaviour, variable-length series, and the distinction between functional channels and semantic labels. It also proposes practical experimental programmes for ECG, machinery monitoring, gait, speech, protein trajectories, and language. The ECG example is treated as an illustrative application rather than an established result. The wider claim is architectural: wherever a measurable sequence can be represented as a trajectory, the TBTmay be investigated as a classifier of dynamical form.

Research Status

This document presents a research architecture and a programme of experiments. It does not claim that all proposed classifier basins have already been demonstrated, nor that any particular application has been clinically or industrially validated. Within the wider FSM and Geofinitist framework, symbolic and measured representations are treated as forming manifolds and basins. The present monograph asks how that general principle may be turned into an operational classifier using the channels and delay-embedded trajectory structure of the TBT.

Contents

Chapter 1

The Classification Insight

1.1 From language trajectory to general trajectory

The Takens-Based Transformer was initially developed in relation to language. A sequence of symbols was treated not as a bag of tokens but as an evolving trajectory. Delay structure supplied the local memory required to reconstruct the path, while support vectors and channels supplied additional functional constraints.

The important extension is that nothing in this general construction requires the sequence to be linguistic. A measurable sequence may be written as

$$X = (x_1, x_2, \dots, x_T),$$

where each x_t may be a scalar, a vector, a symbol, or a registered measurement. The sequence may describe an ECG trace, vibration sensor, spoken waveform, gait sensor, weather record, industrial process, financial series, protein-related sequence, or a language trajectory.

The classification problem may then be restated:

> Given a measured or symbolic trajectory, determine the labelled basin, set of basins, or channel configuration with which that trajectory is most stably associated.

This is a broader role than next-token prediction. The system is no longer required to produce a continuation. It may instead measure, compare, route, classify, reject, or register the trajectory.

1.2 Classification as basin registration

A conventional classifier is commonly presented as a mapping

$$f : \mathcal{X} \to \mathcal{Y},$$

with a final decision

$$\hat{y} = \arg\max_k p(y = k \mid X).$$

The trajectory formulation adds an intermediate geometric object. The measured series is unfolded into a delay representation

$$\Phi_{m,\tau}(X)_t = \left[x_t, x_{t-\tau}, x_{t-2\tau}, \ldots, x_{t-(m-1)\tau}\right],$$

producing a reconstructed trajectory

$$\Gamma_X = \{\Phi_{m,\tau}(X)_t\}_{t=(m-1)\tau+1}^{T}.$$

Classification becomes a comparison between the measured trajectory and one or more learned basin structures:

$$\Gamma_X \longrightarrow \mathcal{B}_k.$$

The class is therefore not treated merely as a final symbol. It is represented by a region of admissible trajectories, a learned channel, a prototype geometry, or a family of stable paths.

1.3 The central thesis

The central thesis of this monograph is:

> The Takens-Based Transformer can be developed as a general trajectory classification architecture in which delay-embedded sequences are registered against one or more labelled channels representing dynamical classes, functional conditions, operational regimes, or measurement distinctions.

This formulation supports both single-label and multi-label systems, but it also permits more expressive arrangements. A channel may describe the final class, a property of the input, a source condition, a quality condition, a trajectory phase, or a routing decision.

Chapter 2

Foundational Terms

2.1 Trajectory

A trajectory is an ordered finite sequence whose present position carries the history required to arrive there. In a measured system, the trajectory is formed through registration over time. In a symbolic system, it is formed through sequential re-registration of marks.

A trajectory may be scalar,

$$x_t \in \mathbb{R},$$

multivariate,

$$\mathbf{x}_t \in \mathbb{R}^d,$$

or symbolic,

$$s_t \in \Sigma.$$

The TBT requires an encoder that places each observation into a common finite representation from which delay relationships can be constructed.

2.2 Manifold

Within this research programme, a manifold is not restricted to an ideal mathematical object detached from measurement. It is a finite, registered organisation of related trajectory positions. The manifold is produced by the chosen measurement, encoding, delay structure, dimensional projection, uncertainty, and historical provenance.

The reconstructed manifold may be written

$$\mathcal{M}_X = \{\mathbf{z}_t\}_{t=1}^{N},$$

where $\mathbf{z}_t$ denotes a learned or constructed trajectory state.

2.3 Basin

A basin is a region in which a family of trajectories remains sufficiently coherent for a particular operational distinction. For classification, a basin is associated with a class or condition. It need not be infinitely smooth, perfectly separated, or physically fundamental. It is an admissible finite organisation that supports reliable distinction under stated measurement conditions.

For class k,

$$\mathcal{B}_k \subseteq \mathcal{M}.$$

A classifier may estimate membership, proximity, attraction, trajectory consistency, or admissibility relative to $\mathcal{B}_k$.

2.4 Channel

A channel is an additional organised dimension supplied to or learned by the model. In language work, channels may guide conversational function. In classification, they may represent labelled distinctions.

A channel can be represented as

$$\mathbf{c}^{(j)} \in \mathbb{R}^{d_c},$$

where j identifies the classification dimension or functional role. A labelled class within that channel may use

$$\mathbf{c}_k^{(j)}.$$

Channels should not be assumed to carry semantic meaning in the ordinary linguistic sense. They organise admissible trajectory structure.

2.5 Support vector

The phrase support vector is used here in the architectural sense developed for the TBT: an auxiliary vector that modifies, partitions, stabilises, or identifies a trajectory. It is not limited to the conventional support vectors of a support-vector machine.

A support vector may encode:

- class identity;
- source identity;
- sensor identity;
- signal quality;
- operating regime;
- trajectory phase;
- energy or completion state;
- uncertainty or admissibility;
- hierarchy level;
- branch selection.

Chapter 3

Single Labelled Channel Classification

3.1 Basic architecture

The simplest classifier uses one labelled channel. Every training trajectory is paired with one class label

$$(X_i, y_i), \qquad y_i \in \{1, \dots, K\}.$$

Each label is associated with a channel representation

$$y_i = k \longleftrightarrow \mathbf{c}_k.$$

The encoder constructs a trajectory representation

$$\Gamma_i = E_\theta(X_i),$$

and the model learns the relationship between Γ_i and the labelled channel basin $\mathcal{B}_k$.

A simple inference rule is

$$\hat{y} = \arg\min_k D(\Gamma_X, \mathcal{B}_k),$$

where D is a learned or constructed trajectory-to-basin distance.

3.2 Three implementation forms

A single labelled channel can be implemented in at least three ways.

3.2.1 Channel-conditioned training

The class channel is supplied during training as an auxiliary signal. The model learns the trajectories admissible under each channel. Care is required to ensure that the model does not merely copy the label. This form is appropriate when the channel is used to construct a generative or reconstruction objective, and the class must still be inferred from the trajectory at deployment.

3.2.2 Learned channel prototypes

Each class has a learned prototype

$$\mathbf{p}_k \in \mathbb{R}^q.$$

The trajectory encoder produces

$$\mathbf{z}_X = g_\theta(\Gamma_X),$$

and classification uses

$$\hat{y} = \arg\min_k \|\mathbf{z}_X - \mathbf{p}_k\|.$$

The prototype may represent a point, a path, a tube, a distribution, or a manifold patch.

3.2.3 Trajectory basin heads

A shared delay-trajectory encoder feeds separate basin heads:

$$h_k(\Gamma_X) \to s_k.$$

The heads may estimate class compatibility, reconstruction quality, stability, or continuation likelihood. The final class may be chosen by maximum compatibility or minimum reconstruction error.

3.3 Diagram of a single-channel classifier

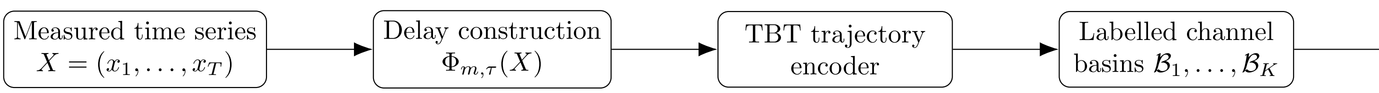

Figure 3.1: Single labelled channel trajectory classification.

3.4 Example applications

A single channel is appropriate when one principal distinction is required. Examples include:

- normal versus anomalous machinery vibration;
- gait category;
- speaker identity;
- rhythm class in an ECG experiment;
- language intent;
- protein family;
- market regime;
- fault type;
- environmental state.

The architecture is unchanged across domains. Only the encoder, sampling assumptions, channel labels, and evaluation criteria differ.

Chapter 4

Multi-Channel Classification

4.1 Why multiple channels matter

Many measured trajectories carry several distinguishable properties at once. An ECG trace, for example, may have a rhythm class, lead identity, signal quality, patient context, acquisition device, and anomaly status. An industrial vibration trace may carry machine identity, load regime, fault type, sensor position, and severity.

A single class label compresses these distinctions into one output. A multi-channel TBT can retain them separately.

Let there be J channels. The label structure is

$$\mathbf{y}_i = \big(y_i^{(1)}, y_i^{(2)}, \dots, y_i^{(J)}\big).$$

Each channel j has its own label set

$$y_i^{(j)} \in \{1, \dots, K_j\}.$$

The full channel state is

$$\mathcal{C}_i = \left(\mathbf{c}^{(1)}_{y_i^{(1)}}, \dots, \mathbf{c}^{(J)}_{y_i^{(J)}}\right).$$

4.2 Parallel multi-channel classification

In the parallel arrangement, a shared trajectory encoder supports several independent classification heads:

$$\Gamma_X = E_\theta(X),$$

$$\hat{y}^{(j)} = h^{(j)}(\Gamma_X), \qquad j = 1, \dots, J.$$

The total loss can be written

$$\mathcal{L}_{\text{parallel}} = \sum_{j=1}^{J} \lambda_j \mathcal{L}_j,$$

where λ_j controls the contribution of each channel.

This architecture is suitable when the channels describe distinct properties that can be estimated from the same trajectory.

4.3 Hierarchical multi-channel classification

In the hierarchical arrangement, an early channel restricts later classification. For example:

$$\text{signal quality} \to \text{signal family} \to \text{specific class}.$$

The model may first classify whether the trajectory is admissible. Only then does it enter a class-specific branch.

$$\Gamma_X \xrightarrow{h^{(1)}} \mathcal{B}_a^{(1)} \xrightarrow{h_a^{(2)}} \mathcal{B}_b^{(2)} \xrightarrow{h_{a,b}^{(3)}} \hat{y}.$$

This is closely related to the earlier observation that explicit filters can form a tree. In a classifier, the tree is no longer merely a textual organisation. It becomes a sequence of learned manifold restrictions.

4.4 Sequential channel filtering

A sequential architecture allows each channel to transform the trajectory representation before the next channel acts:

$$\Gamma^{(0)} = E_\theta(X),$$

$$\Gamma^{(j)} = F_j\big(\Gamma^{(j-1)}, \mathbf{c}^{(j)}\big).$$

The final state

$$\Gamma^{(J)}$$

contains the trajectory after passage through the complete filter chain. This may be useful when later distinctions depend on earlier functional registrations.

4.5 Tree-structured channel classification

A tree permits branching according to channel state:

```
Input trajectory
|
+-- admissible
| |
| +-- normal family
| | +-- subtype A
| | +-- subtype B
| |
| +-- abnormal family
| +-- subtype C
| +-- subtype D
|
+-- inadmissible
   +-- noise
   +-- missing data
   +-- unknown source
```

The tree can be learned, designed, or partially fixed. A designed tree preserves explicit functional distinctions. A learned tree may discover useful partitions but can become difficult to interpret. A hybrid system may be preferable: the high-level channels are explicit, while the local basins are learned.

4.6 Diagram of multi-channel classification

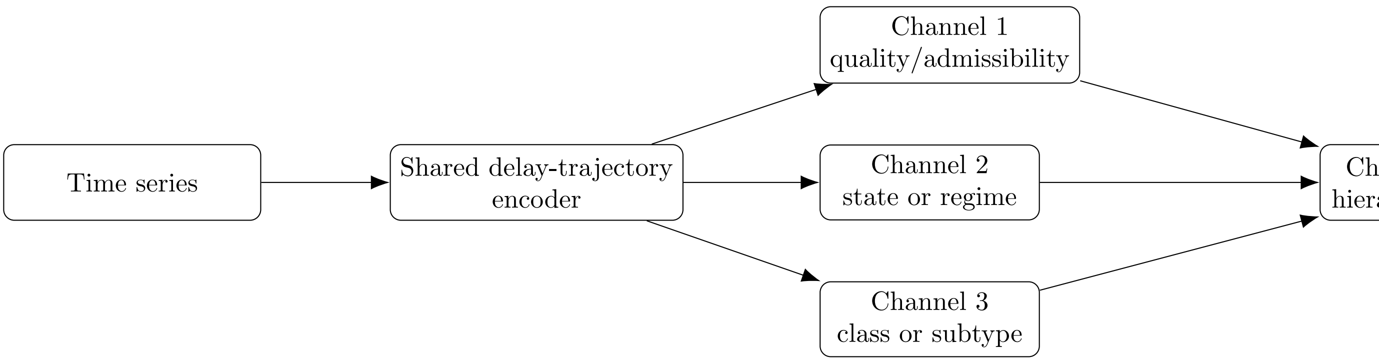

Figure 4.1: A shared trajectory encoder supporting parallel or fused labelled channels.

Chapter 5

Channel Semantics and Functional Separation

5.1 Class channels

A class channel directly represents the target distinction. Examples include normal, atrial fibrillation, ventricular tachycardia, bearing fault, or walking.

5.2 Context channels

A context channel describes the circumstances under which the trajectory was registered. Examples include device, sampling rate, sensor location, patient position, operating load, language, source corpus, or measurement protocol.

Context channels are valuable because two trajectories may appear different for reasons unrelated to the target class. By explicitly representing the measurement context, the model can avoid forcing all variation into the class basin.

5.3 Quality and admissibility channels

A quality channel identifies whether the input is fit for classification. Possible values include

$$\{\text{valid}, \text{noisy}, \text{incomplete}, \text{saturated}, \text{misaligned}, \text{unknown}\}.$$

An admissibility channel may prevent the system from producing a confident class when the measurement does not support one.

5.4 State and phase channels

A measured trajectory may be observed at different phases. A state channel may identify onset, transition, steady regime, recovery, or termination. In language, a similar channel may identify on-ramp, continuation, or end-of-turn energy.

5.5 Identity channels

An identity channel can preserve source-specific information. In speech this may be the speaker; in machinery, the machine; in medicine, the patient or device; in language, the conversational model identity. Identity channels allow the classifier to distinguish within-source dynamics from between-source variation.

5.6 Uncertainty channels

The FSM framework requires uncertainty to be retained rather than silently removed. A channel may therefore encode uncertainty class, interval, provenance strength, or degree of admissibility. A classifier output may be written

$$\hat{Y} = (\hat{y}, \delta, H, A),$$

where δ is uncertainty, H is provenance, and A is admissibility.

Chapter 6

Training the Classifier

6.1 Dataset construction

A training instance should preserve more than an input series and a final label. A richer record is

$$\mathcal{D}_i = (X_i, \mathbf{y}_i, \mathbf{m}_i, H_i, \delta_i, A_i)\,,$$

where

- X_i is the measured trajectory;
- $\mathbf{y}_i$ is the single- or multi-channel label vector;
- $\mathbf{m}_i$ records measurement conditions;
- H_i records provenance;
- δ_i records uncertainty;
- A_i records admissibility.

This prevents the classifier from being trained on labels detached from the conditions under which they were obtained.

6.2 Delay construction

For a scalar series, the delay vector is

$$\mathbf{d}_t = [x_t, x_{t-\tau}, \ldots, x_{t-(m-1)\tau}].$$

For a multivariate series $\mathbf{x}_t \in \mathbb{R}^d$, the delay representation may concatenate channel histories:

$$\mathbf{d}_t = [\mathbf{x}_t, \mathbf{x}_{t-\tau}, \ldots, \mathbf{x}_{t-(m-1)\tau}].$$

Alternatively, separate measurement channels may have separate delay encoders before fusion.

The embedding dimension m and delay τ are finite engineering choices. They must be selected and tested against the actual measured data rather than treated as universal constants.

6.3 Trajectory encoder

The TBT encoder maps the delay sequence to a latent trajectory

$$\mathbf{z}_t = E_\theta(\mathbf{d}_t, \mathbf{s}_t),$$

where $\mathbf{s}_t$ contains support vectors or channels. The complete latent path is

$$Z_X = (\mathbf{z}_1, \ldots, \mathbf{z}_N).$$

The encoder may be trained to preserve local geometry, class separation, reconstruction, continuation, or trajectory smoothness.

6.4 Classification objectives

A practical system may combine several losses.

6.4.1 Channel classification loss

For channel j,

$$\mathcal{L}_{\text{class}}^{(j)} = -\sum_{k=1}^{K_j} y_k^{(j)} \log \hat{p}_k^{(j)}.$$

6.4.2 Prototype loss

$$\mathcal{L}_{\text{proto}} = \|\mathbf{z}_X - \mathbf{p}_y\|^2.$$

6.4.3 Contrastive basin loss

For positive pairs (i, r) from the same basin and negative pairs (i, s) from different basins,

$$\mathcal{L}_{\text{contrast}} = D(\Gamma_i, \Gamma_r) + \max\left(0, \mu - D(\Gamma_i, \Gamma_s)\right).$$

6.4.4 Trajectory reconstruction loss

$$\mathcal{L}_{\text{recon}} = \sum_t \|\hat{x}_t - x_t\|^2.$$

6.4.5 Continuation consistency loss

A trajectory should not change class arbitrarily under a short admissible continuation. One may impose

$$\mathcal{L}_{\text{cont}} = D\left(h(\Gamma_{1:T}), h(\Gamma_{1:T+q})\right)$$

for continuations that remain in the same labelled regime.

6.4.6 Admissibility loss

$$\mathcal{L}_{\text{adm}} = \mathrm{BCE}(a, \hat{a}),$$

where a records whether the sample supports a valid classification.

6.5 Combined objective

A general training objective is

$$\mathcal{L} = \sum_j \lambda_j \mathcal{L}_{\text{class}}^{(j)} + \alpha \mathcal{L}_{\text{proto}} + \beta \mathcal{L}_{\text{contrast}} + \gamma \mathcal{L}_{\text{recon}} + \eta \mathcal{L}_{\text{cont}} + \rho \mathcal{L}_{\text{adm}}.$$

The weights should be treated as experimental controls. Their role is not merely numerical optimisation; they determine which geometrical and functional distinctions the model is encouraged to preserve.

Chapter 7

Inference and Decision Formation

7.1 Complete-trajectory classification

The simplest inference waits for the complete observation window, constructs the delay trajectory, and returns a channel label vector.

$$X_{1:T} \to \Gamma_X \to (\hat{y}^{(1)}, \dots, \hat{y}^{(J)}).$$

7.2 Streaming classification

The TBT is naturally suited to streaming sequences because the delay state can be updated as new measurements arrive. At time t,

$$\Gamma_t = \Gamma_{t-1} \cup \{\mathbf{z}_t\}.$$

The classifier may produce an evolving estimate

$$\hat{y}_t = h(\Gamma_t).$$

A stability rule can prevent premature decisions. For example, a class is registered only if it remains dominant for r successive updates and exceeds an admissibility threshold.

7.3 Trajectory energy and closure

The previously proposed semantic response energy channel has a classifier analogue. A trajectory may carry an evidence or completion energy

$$E_t \geq 0.$$

The system should not necessarily decide while the observed trajectory remains incomplete. A classification may be released when

$$E_t < E_{\min}$$

or when a separate closure channel indicates that sufficient evidence has accumulated.

For streaming signals this may correspond to the end of a beat, event, word, gait cycle, machine transient, or observation window.

7.4 Basin distance and membership

Several decision forms are possible:

$$\begin{aligned}
\text{nearest prototype:} \quad & \hat{y} = \arg\min_k \|\mathbf{z} - \mathbf{p}_k\|, \\
\text{minimum path distance:} \quad & \hat{y} = \arg\min_k D_{\text{path}}(\Gamma, \mathcal{B}_k), \\
\text{maximum compatibility:} \quad & \hat{y} = \arg\max_k s_k(\Gamma), \\
\text{minimum reconstruction error:} \quad & \hat{y} = \arg\min_k \|X - \hat{X}_k\|, \\
\text{channel posterior:} \quad & \hat{y} = \arg\max_k p(c_k \mid \Gamma).
\end{aligned}$$

These are not identical. Each measures a different relation between trajectory and basin. Experiments should compare them rather than assuming a single universal decision rule.

7.5 Unknown and out-of-distribution trajectories

A useful classifier must be able to decline classification. A trajectory may fall between known basins or outside the training manifold. Rejection can be based on

$$\min_k D(\Gamma, \mathcal{B}_k) > \epsilon,$$

low compatibility,

$$\max_k s_k(\Gamma) < \theta,$$

high reconstruction error, unstable channel predictions, or disagreement between parallel channels.

The output should then be

$$\hat{y} = \text{unknown}$$

rather than a forced class.

Chapter 8

Single-Channel versus Multi-Channel Design

8.1 When a single channel is sufficient

A single labelled channel is appropriate when:

- one distinction is operationally sufficient;
- measurement context is tightly controlled;
- source variation is low;
- the classes are mutually exclusive;
- the cost of additional annotation is not justified;
- the first experiment is intended to test whether any basin separation exists.

A single-channel classifier is therefore an excellent minimal viable experiment.

8.2 When multiple channels are necessary

Multiple channels become useful when:

- several independent labels apply to the same trajectory;
- class depends on measurement context;
- quality must be separated from content;
- source identity affects the observed path;
- hierarchical decisions are required;

- classification should be interpretable as a sequence of functional distinctions;
- the same encoder is to support several operational tasks.

8.3 A comparison table

Property	**Single labelled channel**	**Multiple labelled channels**
Primary purpose	One principal class decision	Several properties, stages, or conditions
Annotation burden	Lower	Higher
Interpretability	Simple class basin	Functional decomposition across channels
Robustness to context variation	Limited unless controlled	Context can be represented explicitly
Hierarchical routing	Not inherent	Natural
Unknown detection	One rejection boundary	Can use quality, admissibility, and disagreement
Reuse of encoder	One task	Multiple related tasks
Risk	Over-compression of distinctions	Channel conflict or over-constraint
Best initial use	Minimal proof of concept	Mature domain architecture

Chapter 9

Illustrative ECG Architecture

9.1 Status of the example

The ECG example is used here only to make the architecture concrete. It is not presented as evidence that clinically meaningful ECG classes have already been established as TBT basins, nor as a validated medical system.

Within the broader framework, the working expectation is that registered symbolic and measured trajectories form manifolds and basins. The research question is whether a practical channel-based TBT can learn distinctions useful for classification.

9.2 Single-channel ECG experiment

A minimal experiment could use one lead, one sampling protocol, fixed-duration windows, and a small number of labelled rhythm classes:

$$Y = \{\text{normal}, \text{class A}, \text{class B}, \text{unknown}\}.$$

The pipeline would be

$$\text{ECG window} \to \text{delay embedding} \to \text{TBT trajectory encoder} \to \text{rhythm basin} \to \text{class}.$$

The purpose would be to test whether the delay-embedded TBT representation separates labelled trajectory families better than simple baselines under the same measurement conditions.

9.3 Multi-channel ECG experiment

A richer experiment might use channels such as:

1. lead or sensor channel;
2. signal-quality channel;
3. rhythm-family channel;
4. anomaly channel;
5. uncertainty/admissibility channel.

The output is then not a single diagnosis-like label but a structured registration:

$$\hat{Y} = (\text{lead}, \text{quality}, \text{rhythm family}, \text{anomaly}, \text{admissibility}).$$

This is more consistent with measurement practice because it preserves the conditions under which the final class was produced.

9.4 A possible hierarchical ECG tree

```
ECG trajectory
|
+-- admissible signal
| |
| +-- regular rhythm family
| | +-- class R1
| | +-- class R2
| |
| +-- irregular rhythm family
| +-- class I1
| +-- class I2
|
+-- inadmissible signal
    +-- motion artefact
    +-- poor contact
    +-- missing segment
    +-- unknown acquisition condition
```

The significance of this structure is not its medical completeness. It demonstrates how labelled channels can create functional manifold restrictions before final classification.

Chapter 10

Other Application Domains

10.1 Industrial condition monitoring

A vibration or acoustic series may be classified by machine identity, load, speed, fault family, severity, and sensor position. A multi-channel system can distinguish operational context from fault state.

10.2 Human movement and gait

Accelerometer and gyroscope trajectories can be classified into walking, running, standing, falling, transition, or unknown. Additional channels may represent person identity, sensor placement, terrain, and confidence.

10.3 Speech and acoustic events

Waveform trajectories may be classified by phonetic unit, speaker, emotion, acoustic event, language, or recording quality. A hierarchical system may first identify speech versus non-speech, then language, then speaker or content class.

10.4 Protein and biological sequences

The existing TBT protein work suggests that sequence-derived trajectories may support family, structural, interaction, or regime classification. Here the word "time" should be treated carefully: a sequence index can provide the ordered axis even where the data are not a direct temporal measurement.

10.5 Language classification

Language remains an important application. A trajectory classifier may identify intent, conversational function, topic family, response-energy state, identity basin, question type, or required tool. In this form the classifier becomes a router for a larger language system.

10.6 Financial and operational regimes

Market or process data may be classified into regimes. The principal caution is that these systems are non-stationary and reflexive. Channels should preserve date, market, instrument, sampling method, and provenance rather than treating all historical trajectories as equivalent.

Chapter 11

Relationship to Filtering and Routing

11.1 Filters as classifiers

The earlier filtering experiment showed that explicit symbolic markers can create nested functional regions. In a trained TBT, a channel can perform the same role numerically.

A filter asks:

> Which trajectories remain admissible after this distinction?

A classifier asks:

> Within which labelled region does this trajectory belong?

These are closely related operations. A hierarchical classifier may be understood as a tree of manifold filters.

11.2 Classification as routing

A class decision can route the trajectory to a specialist model:

$$X \to R(X) \to M_{R(X)}.$$

The router itself may be a small TBT classifier. This permits an architecture composed of specialised low-compute models rather than one universal model.

11.3 Classification as functional decomposition

Multiple channels can decompose a complex decision:

$$\text{input} \to \text{quality} \to \text{source} \to \text{state} \to \text{class} \to \text{action.}$$

The system therefore preserves the trajectory of the decision rather than compressing it into a single opaque output.

Chapter 12

Experimental Programme

12.1 Stage 1: minimal single-channel proof of concept

The first experiment should use:

- one domain;
- one controlled input format;
- two to five classes;
- one labelled channel;
- fixed training and validation splits;
- a small conventional baseline;
- explicit logging of delay parameters and model state.

The objective is not to achieve a headline result. It is to determine whether stable class separation emerges in the trajectory representation.

12.2 Stage 2: basin inspection

The latent trajectories should be inspected for:

- within-class coherence;
- between-class separation;
- crossover regions;
- shortest-path misrouting;
- sensitivity to delay τ;

- sensitivity to embedding dimension m;
- stability under noise;
- stability under truncated windows;
- effect of repeated training examples;
- dependence on source identity.

12.3 Stage 3: add a quality channel

The first additional channel should normally be quality or admissibility. This tests whether the model can distinguish "unknown because it belongs elsewhere" from "unknown because the measurement is poor."

12.4 Stage 4: add a context channel

A source, sensor, device, or operating-condition channel can then be added. The experiment should ask whether explicit context reduces distortion of the principal class basin.

12.5 Stage 5: hierarchical routing

A high-level channel routes the trajectory to a specialist classifier. This tests whether a tree of smaller models performs better or more transparently than a single multi-class head.

12.6 Stage 6: streaming inference

The classifier is then tested on partial trajectories. The main measurements are time-to-stable-class, early error rate, energy-to-closure behaviour, and correction after misleading initial segments.

12.7 Stage 7: unknown trajectories

Out-of-class, corrupted, and synthetic crossover examples should be included. The system should be rewarded for refusal where no trained basin is admissible.

Chapter 13

Evaluation

13.1 Conventional metrics

Useful conventional metrics include accuracy, balanced accuracy, precision, recall, F1 score, confusion matrices, receiver-operating characteristics, calibration, and Brier score. These remain useful but do not fully describe trajectory behaviour.

13.2 Trajectory-specific metrics

Additional metrics should include:

- basin compactness;
- inter-basin separation;
- trajectory crossover frequency;
- classification stability over successive windows;
- time to stable registration;
- channel disagreement rate;
- unknown rejection accuracy;
- perturbation sensitivity;
- continuity under admissible extension;
- path-length distortion;
- dependence on provenance or source.

13.3 A basin separation measure

For class basin centroids $\boldsymbol{\mu}_k$ and within-basin dispersion σ_k, a simple exploratory score is

$$S_{ij} = \frac{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|}{\sigma_i + \sigma_j + \varepsilon}.$$

This is not a complete manifold measure, but it gives a first indication of whether the encoded classes occupy distinguishable regions.

13.4 Trajectory consistency

For successive windows $X_{1:t}$ and $X_{1:t+q}$ from the same event, consistency may be measured by

$$C_t = 1 - D\big(h(\Gamma_{1:t}), h(\Gamma_{1:t+q})\big).$$

A stable classifier should increase consistency as the relevant trajectory becomes complete.

Chapter 14

Failure Modes and Design Risks

14.1 Label copying

If the true class channel is supplied directly to the model during training without a suitable objective, the model may learn to reproduce the label rather than infer it from the trajectory. Class channels should therefore be used as targets, prototypes, contrastive organisers, or reconstruction conditions with carefully separated inference pathways.

14.2 Channel leakage

A context channel may accidentally reveal the target class. Dataset construction should test for this. For example, if one device recorded only one class, the device channel becomes a surrogate label.

14.3 Over-partitioning

Too many channels may fragment the training space so severely that each basin contains too few trajectories. Channels should represent operationally meaningful distinctions, not every available metadata field.

14.4 Basin collapse

Different classes may collapse into the same latent region. This may arise from inadequate delay structure, insufficient encoder capacity, poor labels, or genuinely indistinguishable measurements.

14.5 False geometric confidence

A visually separated low-dimensional projection does not by itself establish a reliable classifier. Separation must be tested under held-out data, source changes, noise, and adversarial or crossover cases.

14.6 Non-stationarity

The learned manifold may drift as equipment, patients, language, markets, or sensors change. Provenance and date must therefore remain part of the registered classification system.

14.7 Forced classification

A classifier that always returns one of the known classes hides uncertainty. Unknown, inadmissible, and insufficient-evidence outputs should be first-class channels.

Chapter 15

Implementation Sketch

15.1 Data record

A practical record might be represented as:

```
record = {
    "series": X, # shape: [time, features]
    "labels": {
        "class": class_id,
        "quality": quality_id,
        "source": source_id,
        "state": state_id,
    },
    "sampling_rate": sampling_rate,
    "provenance": provenance,
    "uncertainty": uncertainty,
    "admissible": admissible,
}
```

15.2 Model sketch

```
class TBTClassifier:
    def __init__(self, delay_encoder, trajectory_encoder, channel_heads):
        self.delay_encoder = delay_encoder
        self.trajectory_encoder = trajectory_encoder
        self.channel_heads = channel_heads

    def forward(self, series, support_vectors=None):
        delays = self.delay_encoder(series)
        trajectory = self.trajectory_encoder(delays, support_vectors)
        outputs = {
            name: head(trajectory)
            for name, head in self.channel_heads.items()
        }
        return trajectory, outputs
```

15.3 Training sketch

```
for batch in loader:
    trajectory, outputs = model(
        batch["series"],
        support_vectors=batch.get("support_vectors")
    )

    loss = 0.0
    for channel_name, prediction in outputs.items():
        target = batch["labels"][channel_name]
        loss += channel_weights[channel_name] * criterion(
            prediction, target
        )

    loss += prototype_weight * prototype_loss(
        trajectory, batch["labels"]["class"]
    )

    loss += admissibility_weight * admissibility_loss(
        outputs["admissibility"], batch["admissible"]
    )

    optimiser.zero_grad()
    loss.backward()
    optimiser.step()
```

15.4 Inference sketch

```
def classify(series):
    trajectory, outputs = model(series)

    if outputs["admissibility"].score < ADMISSIBILITY_THRESHOLD:
        return {"class": "unknown", "reason": "inadmissible"}

    result = {
        channel: prediction.best_label
        for channel, prediction in outputs.items()
    }
    result["trajectory"] = trajectory
    return result
```

Chapter 16

Development Platform Requirements

A TBT classification workbench should make the trajectory and channel structure visible. A minimal interface should include:

- dataset and provenance panel;
- signal viewer;
- delay-embedding controls;
- encoder selection;
- channel-definition editor;
- single-channel or multi-channel mode;
- hierarchy/tree editor;
- training parameters;
- live loss and channel metrics;
- trajectory projection viewer;
- basin inspection;
- crossover and misrouting inspection;
- unknown/rejection tests;
- model and interface state saving;
- experiment logs;
- export of trained artefacts and metadata.

The interface should be able to restore a previous experimental position. The channel tree, delay parameters, dataset version, training state, and visualisation state should therefore be included in a meta-dataset describing the experiment itself.

Chapter 17

Implications for the TBT Programme

17.1 Beyond a language model

The classifier architecture changes the strategic position of the TBT. It need not be judged only against large attention-based language models. It can be developed as a compact trajectory instrument for domains where temporal or ordered structure matters.

17.2 A family of specialised instruments

Rather than one universal model, the architecture supports a family of small specialist classifiers connected by channels and routing:

$$\text{general input router} \to \text{domain classifier} \to \text{specialist basin model} \to \text{decision or tool.}$$

This aligns with the wider objective of building systems that can operate on modest local hardware.

17.3 Classification before generation

A language system may first classify the trajectory before generating a response. It may determine conversational function, identity channel, energy requirement, tool need, or domain. Generation then occurs within the selected functional manifold.

17.4 A bridge between measurement and symbolic action

The classifier receives registered measurements, constructs a finite trajectory, associates that trajectory with labelled basins, and returns an operational symbol or action. It therefore occupies the generonic boundary between measurement and symbolic response.

Chapter 18

Open Questions

The following questions define the immediate research frontier:

1. What is the simplest delay-embedded TBT classifier that demonstrates stable basin separation?
2. Should a class basin be represented by a point, distribution, path, tube, graph, or specialist decoder?
3. Which channel types improve classification and which merely fragment the data?
4. How should channel interactions be represented when labels are dependent?
5. Can a channel tree be learned without losing interpretability?
6. How should a trajectory be rejected when it lies between known basins?
7. Can energy channels determine when enough of a sequence has been observed?
8. How well does the classifier transfer across sensors, people, machines, or corpora?
9. Can short trajectories be mapped reliably to long-duration basin structure?
10. How should memory half-lives and rotating windows alter streaming classification?
11. Can chained TBT models retrain lower-memory layers while the upper classifier remains operational?
12. Can classifier channels be converted directly into routing channels for tools and specialist models?
13. What metrics best capture trajectory fidelity rather than only final label accuracy?
14. How can provenance and uncertainty remain attached to every classification?

Chapter 19

Conclusion

The essential insight developed in this monograph is that the channel architecture of the Takens-Based Transformer can be used to construct a general classifier of ordered and time-dependent data.

In the simplest form, one labelled channel defines a set of class basins. A measured sequence is delay-embedded, encoded as a trajectory, and registered against those basins. In the richer form, several channels classify different aspects of the same trajectory: quality, context, identity, state, class, uncertainty, and admissibility. These channels may operate in parallel, in sequence, or as a branching tree of manifold filters.

This shifts classification away from a single compressed output and towards an explicit trajectory of distinctions. The final class can be accompanied by the conditions and functional path through which it was reached.

The ECG example illustrates one possible use, but it is not the foundation of the claim. The broader proposition is that any measured or symbolic sequence capable of finite ordered registration may be investigated through the same architecture. The TBT therefore becomes not only a candidate language model, but a general delay-embedded trajectory classification system.

The immediate next step is experimental and modest: construct a minimal single-channel classifier, inspect whether stable labelled basins emerge, and then add channels one at a time. The strength of the idea will not come from declaration, but from whether these channels produce measurable, reproducible, and operational distinctions across real trajectories.

Keywords

Takens-Based Transformer; Functional Symbolic Mechanics; Geofinitism; time-series classification; trajectory classification; delay embedding; channel architecture; support vectors; manifold filters; classifier basins; multi-channel learning; hierarchical classification; streaming classification; admissibility; uncertainty; provenance.