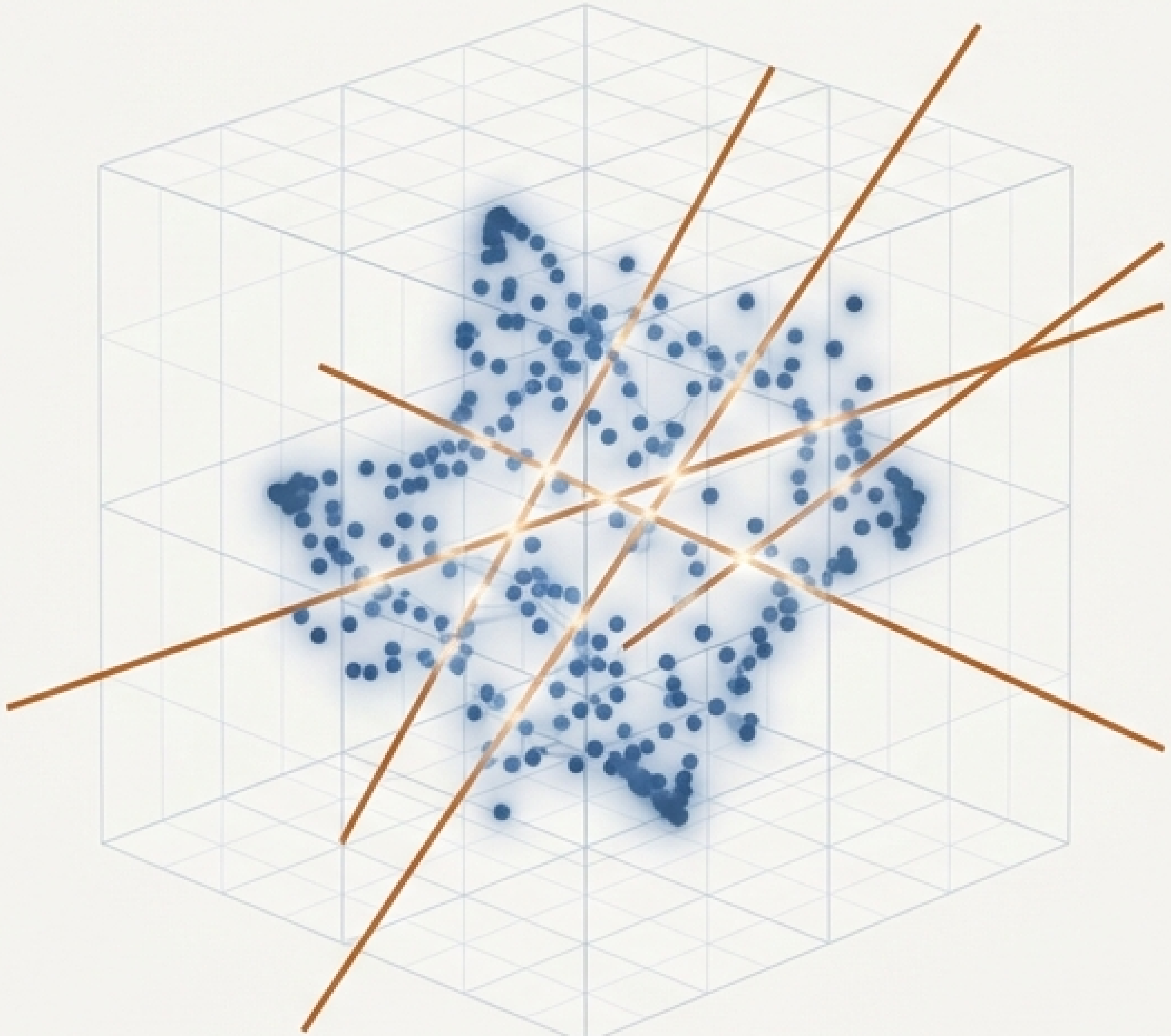
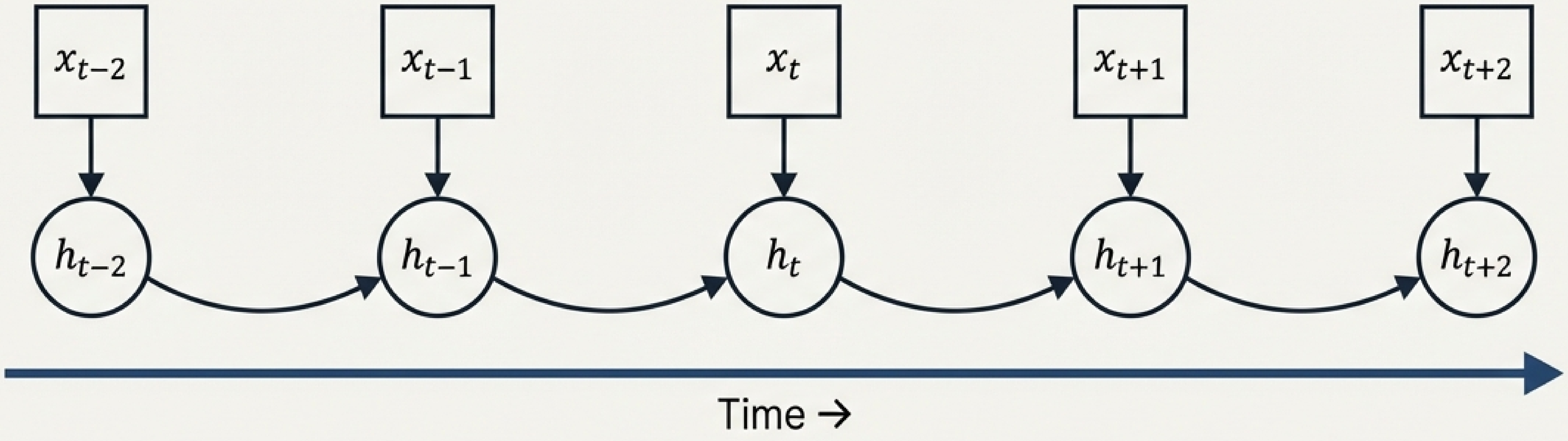
Rethinking Order: Recurrent Neural Networks Beyond Time

Introducing Structural Evolution RNNs (SE-RNNs)



Vagan Terziyan, Artur Terziian, Oleksandra Vitko

The Conventional View: Recurrence is Fundamentally Tied to Time



For decades, the power of RNNs has been almost exclusively harnessed for temporal sequences, conflating the model's mechanism (order) **with its most common application** (time).

Canonical Use Cases

- Natural language processing
- Speech recognition
- Time-series forecasting

Implicit Assumption

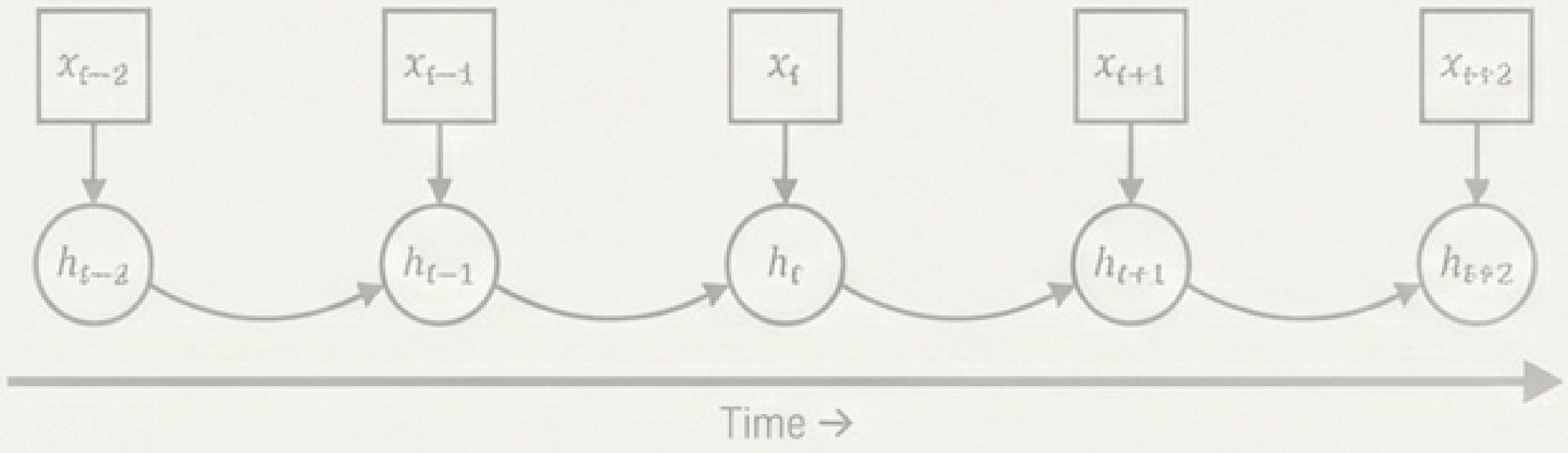
The sequence index *is* time. This has become deeply embedded in both the conceptual understanding and practical use of recurrent architectures.

The Limitation

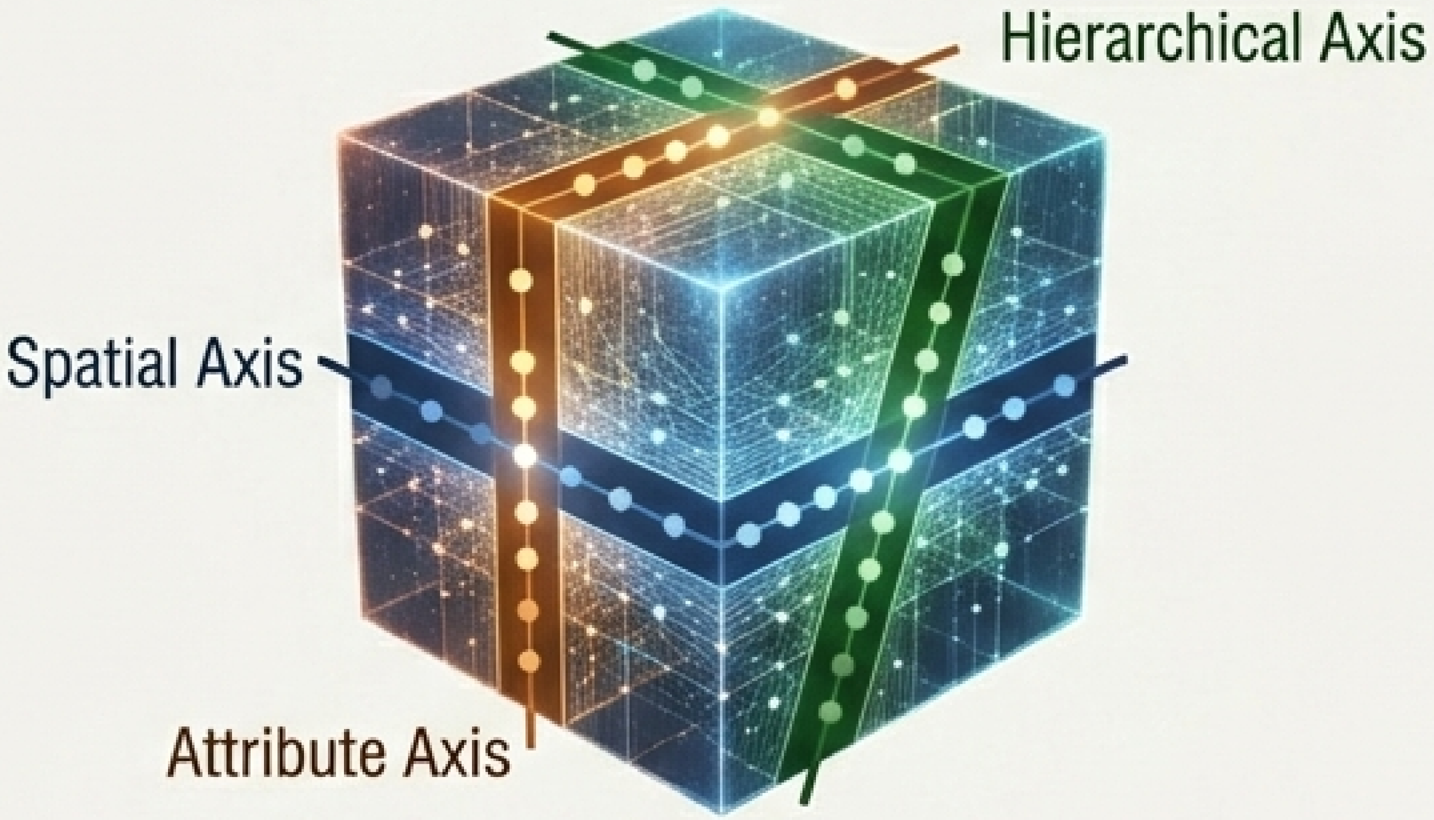
This "time-centric" view discards rich, non-temporal structural information present in many complex datasets.

The Real Insight: RNNs are Models of Ordered Data, Not Time Itself

Conventional View



Proposed View



The core recurrence relation $h_t = f(h_{t-1}, x_t)$ operates on an ordered index, not a physical clock. This opens the door to modeling any meaningful sequence.

What an RNN Actually Models

A function over a totally ordered index set.
An assumption that a recursive state can summarize prior structure.

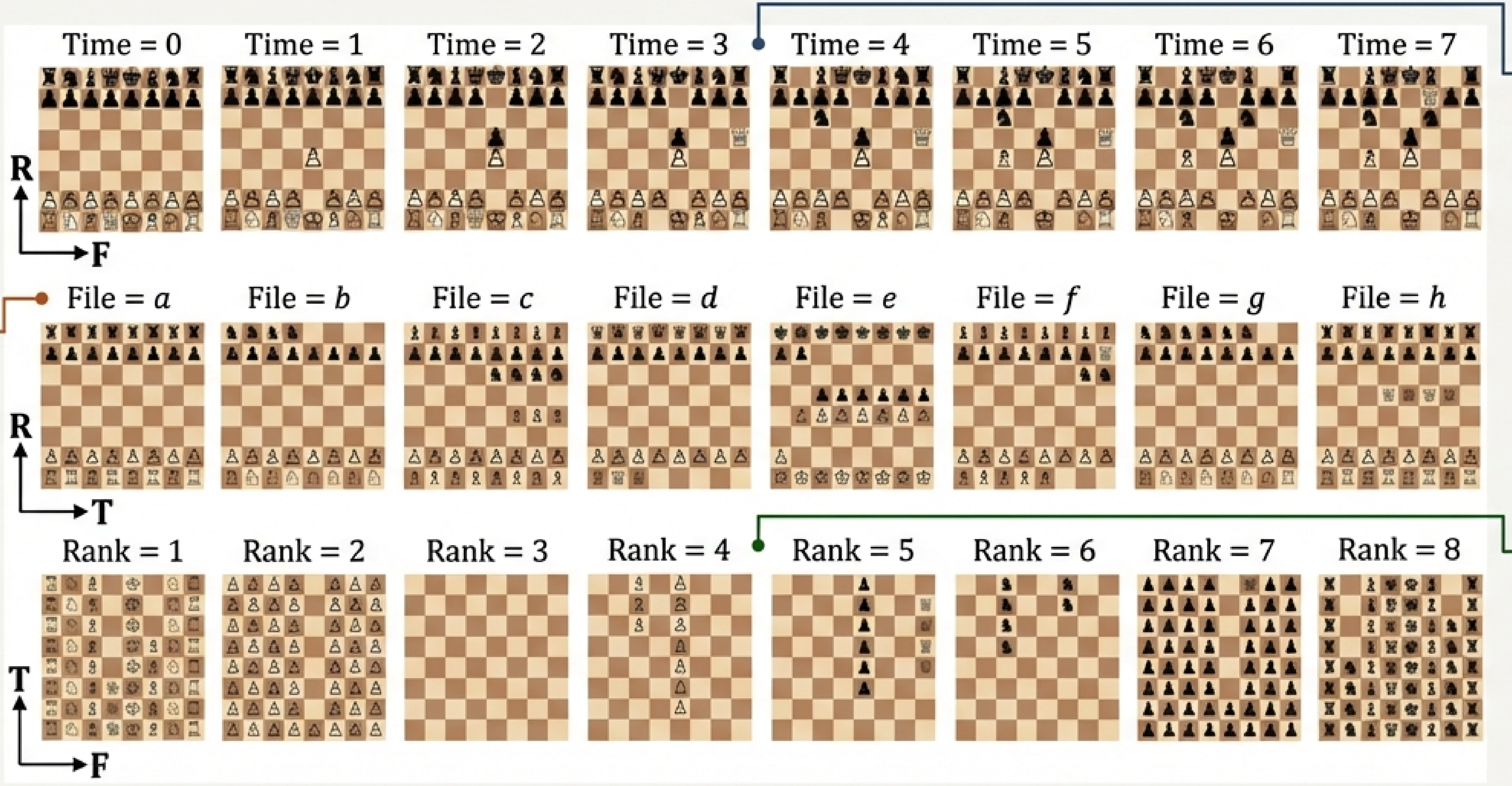
> "The defining feature of an RNN is recurrence over an index, not time."

What an RNN Does NOT Intrinsically Model

Duration, time intervals, or irregular gaps.
Simultaneity or asynchronous events.
Physical causality (this is imposed by the data's ordering, not the model).

A Chess Game Isn't One Story, It's Three: Time, Files, and Ranks

The same chess game can be viewed as multiple, distinct stories (sequences) by 'slicing' the data cube along different axes. Each view reveals unique strategic patterns.



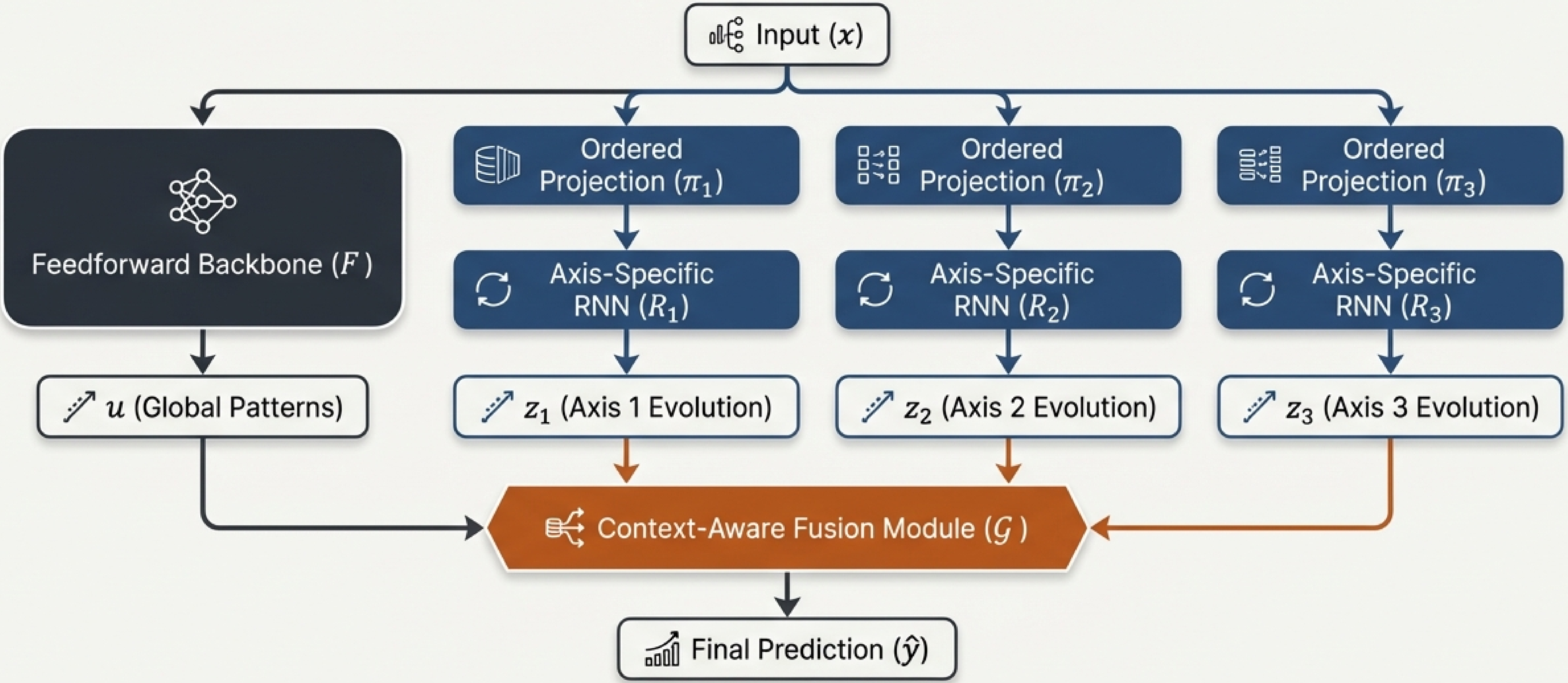
Time Slices (The Obvious Story): A sequence of board states, capturing move-by-move dynamics and classical sequential move dynamics.

File Slices (The Vertical Story): Sequences of how each column evolves, revealing vertical control, pawn structure, and sequences of vertical interactions.

Rank Slices (The Horizontal Story): Sequences of how each row evolves, revealing horizontal control, defensive lines, and sequences of horizontal interactions.

We Built an Architecture to Learn from Multiple Structural Views Simultaneously

Structural Evolution RNNs (SE-RNNs) integrate a Feedforward Network (for global patterns) with multiple specialized RNNs (for structural "views") via a smart fusion module.



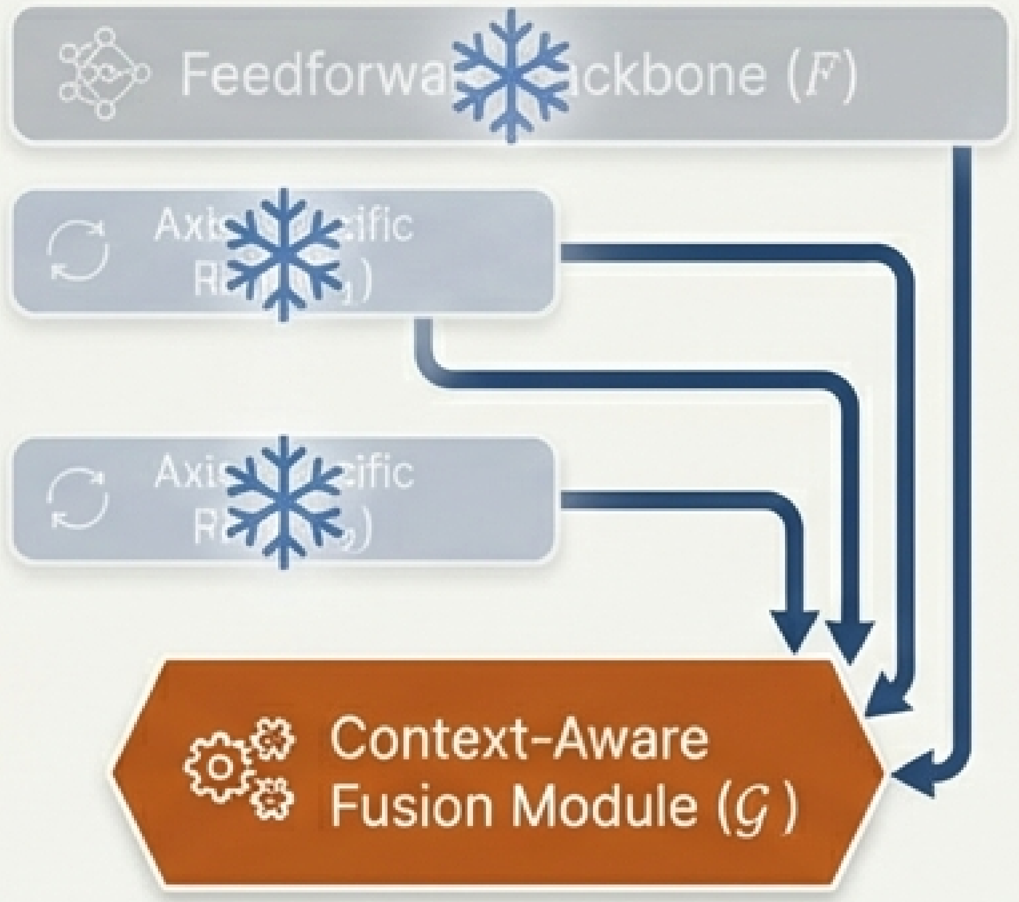
A Modular Training Strategy Ensures Specialization and Stability

Components are trained independently first to become ‘experts,’ then a fusion module learns how to best combine their insights, avoiding interference and boosting efficiency.

1 Pre-train in Parallel

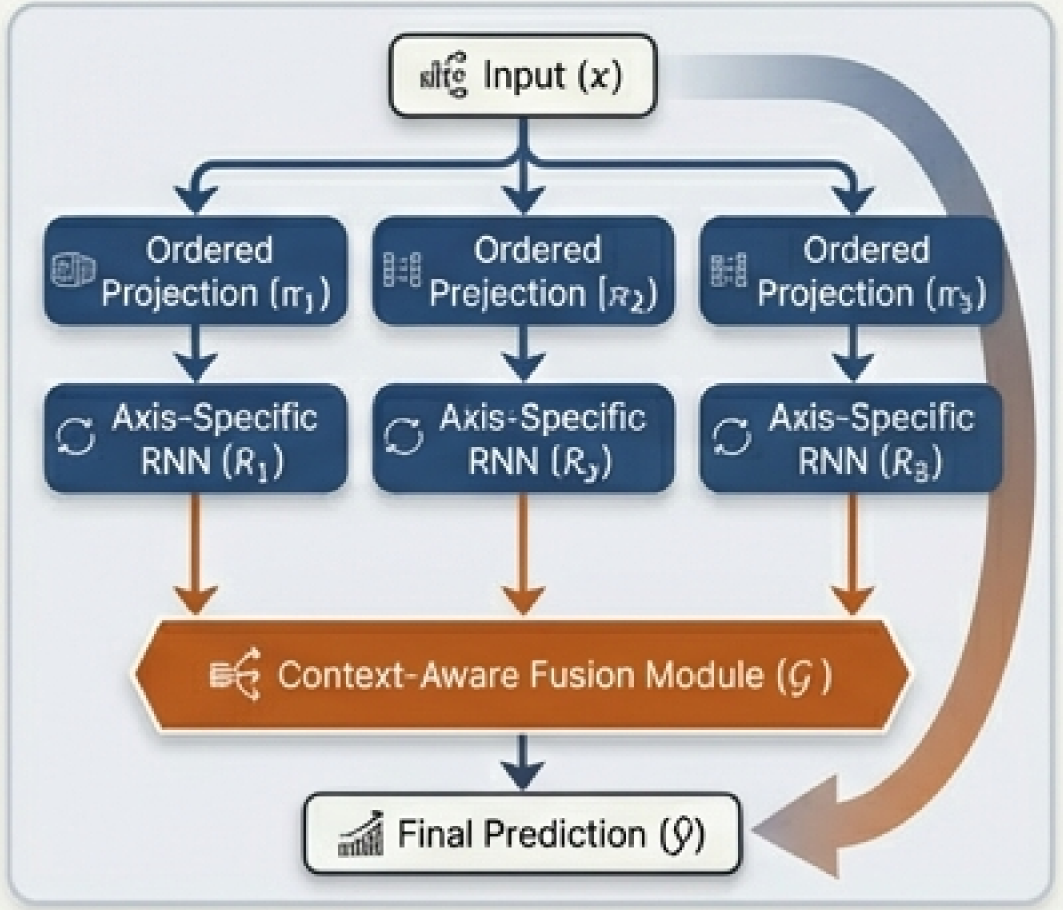
Train the Feedforward Backbone and all Axis-RNNs independently. This stage enforces axis-specific specialization and is highly efficient.

2 Freeze & Train Fusion



Freeze the weights of the pre-trained “experts.” Train only the Fusion Module to learn how to integrate their specialized signals.

3 (Optional) Fine-Tune



Perform a final, gentle end-to-end fine-tuning of the entire system with a low learning rate for final adjustments.

We Designed Datasets to Test SE-RNNs Under Varying Structural Complexity

To validate our architecture, we created three datasets with distinct characteristics to test where the SE-RNN provides the most value.

Dataset 1 (Moderate Complexity)

$$y = \sin(x_1 \cdot x_2) + \cos(x_2 \cdot x_3) + \tanh(x_1 - x_3) + \varepsilon$$

Tests the ability to capture smooth, predictable structural evolution.

Dataset 2 (High Complexity)

$$y = \cos(x_1) \cdot x_2 - \sin(x_2 \cdot x_3) + 0.1 \cdot \varepsilon$$

Tests robustness on chaotic, multiplicative-oscillatory, and entangled relationships.

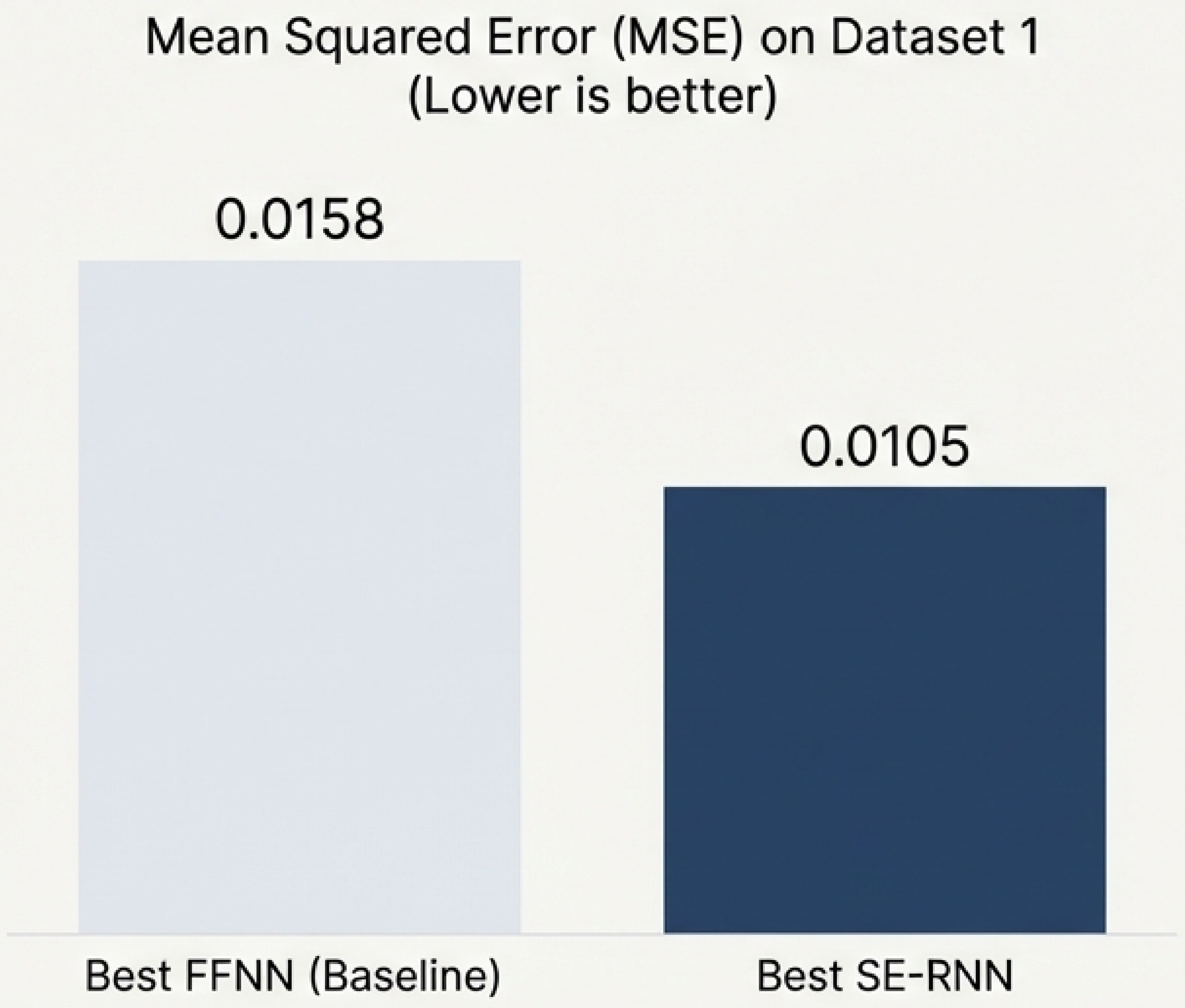
Dataset 3 (Adversarial/Simple)

$$y = 1.5x_1 - 2.0x_2 + 0.5x_3 + \varepsilon$$

Tests whether the architecture adds unnecessary noise when no complex structure exists.

On Data with Clear Hidden Structure, SE-RNNs Consistently Outperform the Baseline

On Dataset 1, multi-axis RNNs capture structural patterns that the feedforward network misses, leading to significant performance gains.



Against Weak Baseline

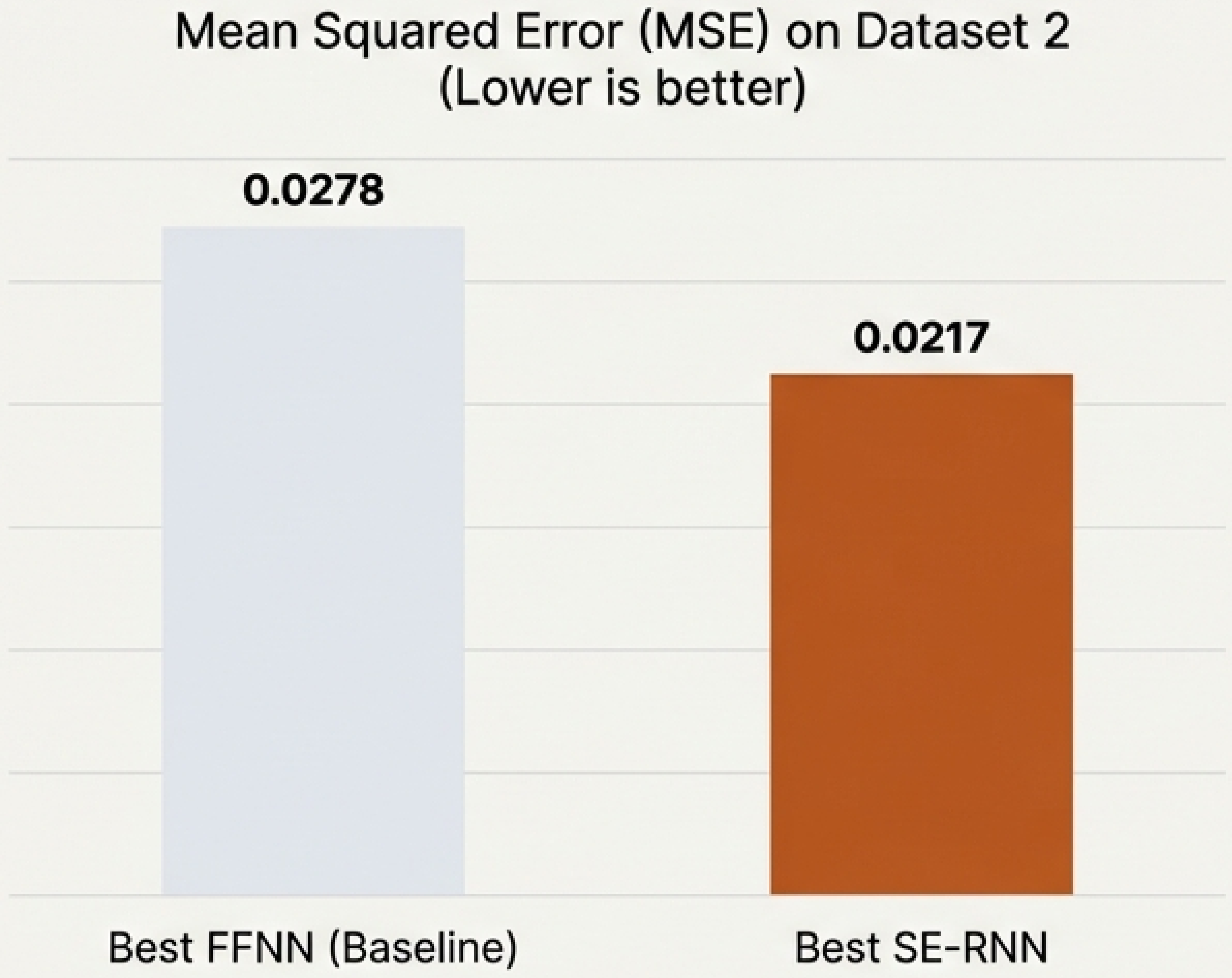
With strong RNNs and fusion, SE-RNN achieved a **72.4%** MSE reduction (Run 6).

Against Strong Baseline

Even against a strong FFNN, the full SE-RNN architecture improves performance by **30.0%** (Run 8).

As Data Complexity Increases, the Value of Structural Projections Becomes Even More Critical

On the highly complex Dataset 2, the SE-RNN advantage over a purely feedforward approach is even more pronounced.



Mastery over Complexity

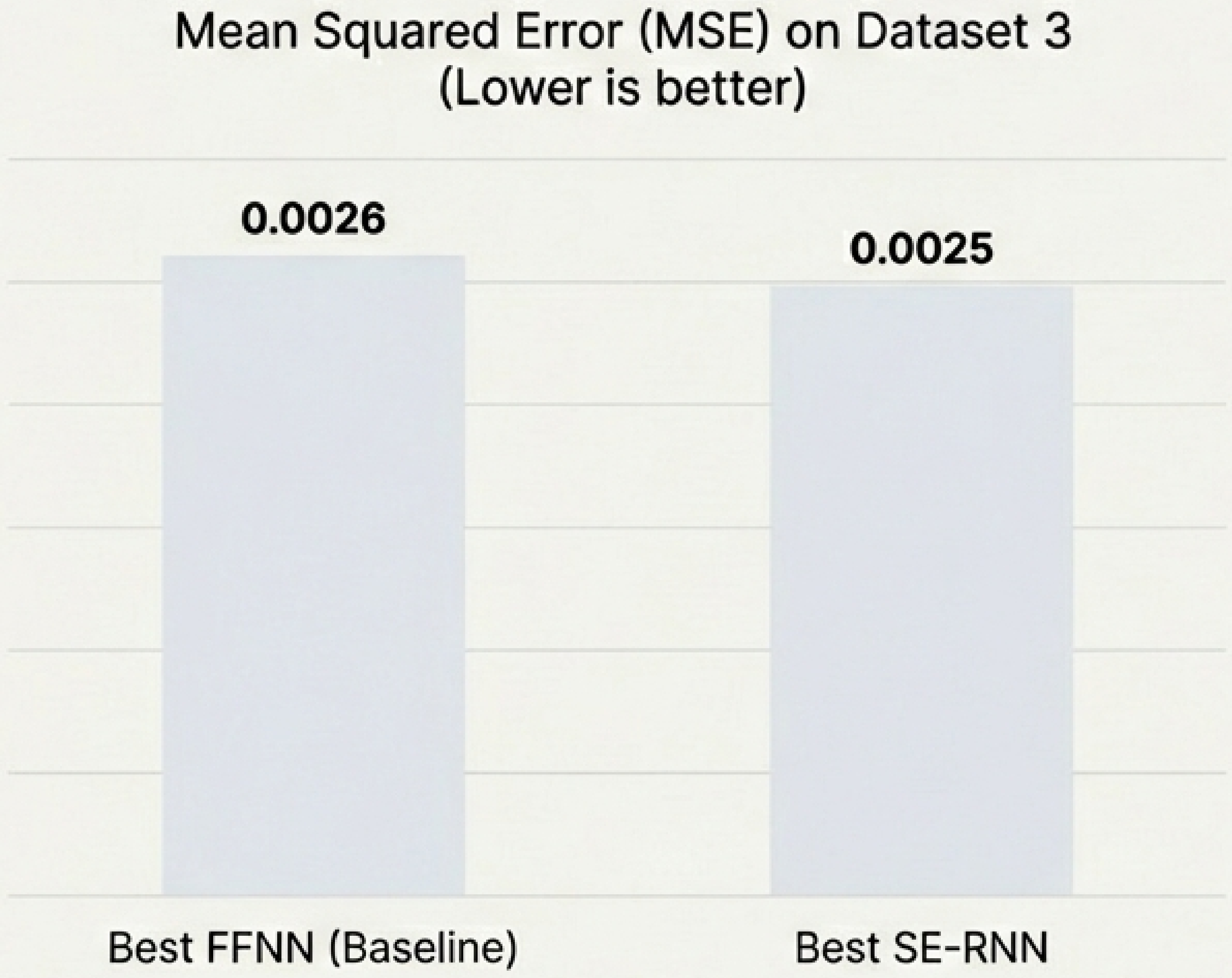
A combination of strong RNNs and fusion achieved a **92.4% reduction in error** over a weak baseline (Run 6).

Value with Strong Baseline

The fully-trained SE-RNN still improves on a very strong FFNN baseline by **21.0%** (Run 8).

In Adversarial Scenarios, SE-RNNs Are a ‘Safe’ Choice, Gracefully Matching the Baseline

When no complex hidden structure exists, the SE-RNN architecture robustly performs on par with the simpler, sufficient feedforward model.



Comparable Performance

The best SE-RNN configuration (MSE: 0.0025) performs comparably to—and even slightly better than—the best FFNN baseline (MSE: 0.0026).

Low-Risk, High-Reward

The added complexity introduces minimal overhead. You either get a significant performance boost on complex data or you match the baseline on simple data.

The Advantage of SE-RNNs Scales Directly with the Data's **Hidden Structural Complexity**

The experimental results show a clear pattern: the richer the hidden relationships in the data, the more benefit SE-RNNs provide.

SE-RNN ≈ FFNN.
Benefit: Robustness.

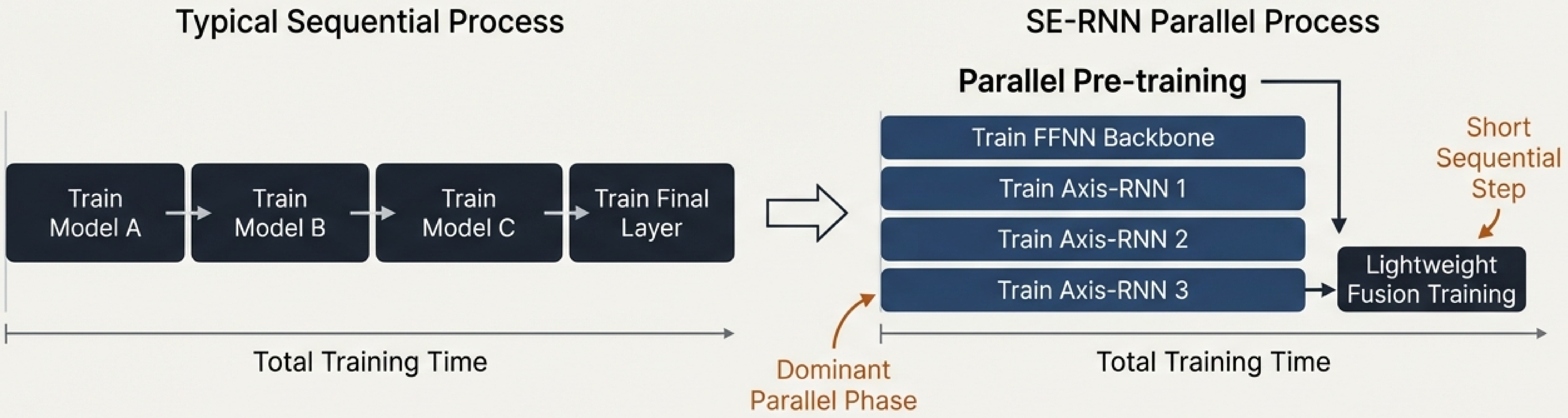
SE-RNN >> FFNN.
Benefit: Significant Improvement.

SE-RNN >>> FFNN.
Benefit: Critical Advantage.

Hidden Structural Complexity →

The Modular Design Allows for Highly Parallel and Efficient Training

SE-RNNs are not computationally prohibitive. Most of the heavy lifting is done in parallel, making training time largely independent of the number of axes.



- **Parallel Pre-training**: The FFNN and all Axis-RNNs can be trained simultaneously. Training time does not scale linearly with the number of axes.
- **Lightweight Fusion**: The only sequential step is training the relatively small fusion module on the frozen outputs of the experts.

SE-RNNs Occupy a Unique Niche in Structural and Sequence Learning

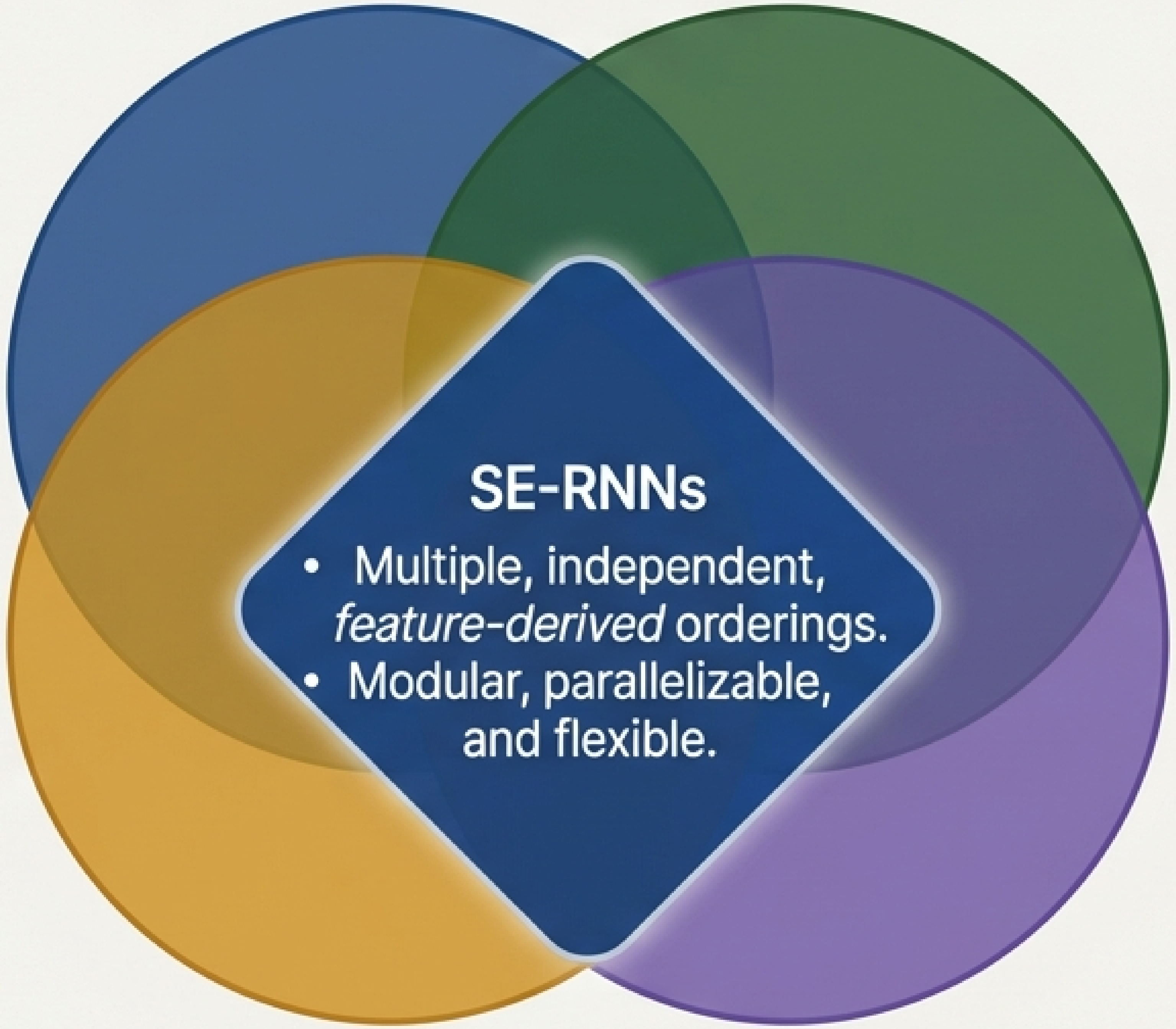
SE-RNNs combine ideas from multiple fields but create a new category focused on internally-derived structural projections, not external modalities or predefined graphs.

Standard RNNs/LSTMs

- Single-axis, typically temporal.

Graph Neural Networks (GNNs)

- Require an explicit, pre-defined graph structure.



Multi-View Learning

- Views are typically external modalities (image, text), not generated from internal structure.

Multi-Dimensional RNNs (MDRNNs)

- Tightly-coupled recurrence on grids; not modular or parallel.

SE-RNNs Offer a More General and Powerful Framework for Recurrent Modeling

By decoupling recurrence from time, SE-RNNs unlock the potential to model rich structural evolution along any meaningful axis in the data.

1 Conceptual Contribution

Reframed RNNs as models of **ordered structural evolution**, with time as just one special case. This disentangles the core mechanism of recurrence from its most common application.

2 Architectural Contribution

Proposed a novel, modular, and efficient architecture (SE-RNN) that integrates multi-axis recurrent models with a feedforward backbone via a context-aware fusion module.

3 Empirical Contribution

Demonstrated that the **SE-RNN advantage scales with data complexity**, offering significant gains on structured data while remaining robust and safe on simple data.

The Road Ahead: Exploring Learned Projections and Real-World Applications

This work opens up exciting avenues for future research at the intersection of structure, sequence, and learning.

Future Work

- Applying SE-RNNs to real-world structured data (e.g., spatial-temporal analytics, tabular data with known hierarchies).
- Developing methods to automatically *learn* the most informative projections instead of defining them manually.
- Exploring more advanced fusion mechanisms , such as attention-based or probabilistic integrators.

Thank you.

Code and datasets available at:

https://ai.it.jyu.fi/experiments/SE-RNNs/

Contact Information:

Vagan Terziyan
vagan.terziyan@jyu.fi