Binding by Oscillatory Dynamics in Neural Architectures for Relational Reasoning

Elder G. Veliz
Advisor: John D. Lafferty

Submitted to the faculty of the Department of Statistics & Data Science
in partial fulfillment of the requirements for the degree of Bachelor of Science

Department of Statistics & Data Science
Yale University
MAY 1ST, 2025

Abstract

Relational reasoning requires a learning system to represent *how* entities are related, not merely *what* they are. Inspired by neuroscientific evidence that synchronous neural oscillations dynamically bind distributed features, we propose a differentiable Kuramoto-based oscillator network that embeds binding-by-synchrony into modern deep learning. Each feature dimension of two input objects is realized as a pair of coupled phase oscillators whose interactions are governed by learned within- and between-object coupling tensors. After a short roll-out of vector-valued Kuramoto dynamics, cross-object *phase coherence* forms a compact *relational bottleneck* that is fed to a lightweight classifier.

We evaluate the model on two canonical benchmarks of abstract reasoning—the *same/different* discrimination task and the *relational match-to-sample* (RMTS) task—under three increasingly difficult generalization regimes. Across five random seeds, the oscillator architecture matches or exceeds the accuracy of a CNN+MLP baseline while converging faster in transfer learning and sustaining a consistent advantage on test accuracy. Ablation studies pinpoint the *between-object coupling matrix* as the critical inductive bias: removing this term largely eliminates both the training-speed gain and the generalization margin. Energy-landscape analysis confirms that, when couplings are symmetric and intrinsic frequencies are ablated, the network descends a Lyapunov energy function whose minima separate "same'' from "different'' inputs. Unified phase portraits further show that training sculpts the oscillator manifold so that identical inputs trigger global phase locking, whereas non-identical inputs fragment into partially synchronous sub-assemblies.

Together, these results demonstrate that biologically plausible oscillatory dynamics can endow deep networks with interpretable, data-efficient relational abstraction. The work opens a path toward integrating dynamical binding mechanisms with high-capacity architectures for complex reasoning.

Table of contents

1 Introduction **4**
- 1.1 Relational Reasoning and the Binding Problem . . . 4
- 1.2 Symbolic vs. Neural Approaches to Relational Reasoning . . . 4
- 1.3 The Relational Bottleneck Principle . . . 4
- 1.4 Neural Oscillations for Binding . . . 5
- 1.5 Kuramoto Oscillator Networks in Deep Learning . . . 5
- 1.6 Our Contribution . . . 5

2 Approach **6**
- 2.1 Oscillatory Dynamics as a Binding Mechanism . . . 6
- 2.2 Coherence as a Relational Bottleneck . . . 6
- 2.3 Two-Stage Transfer Learning Framework . . . 6
- 2.4 Comparison with Alternative Binding Approaches . . . 7

3 Model Architecture **7**
- 3.1 Notation . . . 8
- 3.2 Feature Extraction . . . 8
- 3.3 Kuramoto Oscillator Network . . . 9
  - 3.3.1 Oscillator Representation . . . 9
  - 3.3.2 Kuramoto Dynamics . . . 9
  - 3.3.3 Update Procedure . . . 10
- 3.4 Energy Function . . . 10
- 3.5 Coherence Measurement . . . 11
- 3.6 Classification Components . . . 11
  - 3.6.1 Same/Different Classification . . . 11
  - 3.6.2 RMTS Classification . . . 12

4 Tasks **12**
- 4.1 Visual Stimuli . . . 12
- 4.2 Same/Different . . . 12
- 4.3 Relational Match-to-Sample (RMTS) . . . 13
- 4.4 Generalization Regimes . . . 14

5 Training Details **14**
- 5.1 Model Configuration . . . 14
- 5.2 Optimization . . . 14
- 5.3 Training Procedure . . . 15
- 5.4 Data Processing . . . 15
- 5.5 Alternative Architectures . . . 15
  - 5.5.1 CNN+MLP Baseline . . . 15
  - 5.5.2 Ablated Oscillator Models . . . 15

6 Results and Discussion **16**
- 6.1 Training (In-Distribution) Performance and Convergence . . . 16
- 6.2 Generalization (Out-of-Distribution) Performance . . . 17
- 6.3 Evolution of Energy Landscape . . . 18
- 6.4 Phase–Space Visualization . . . 19
- 6.5 Unified Phase Portrait . . . 21

7 Conclusion **23**

8 Limitations and Future Directions **23**

A Code Availability **26**

Acknowledgments

I wish to express my deepest gratitude to Professor John D. Lafferty—a machine-learning luminary, visionary scientist, and, above all, an inspiring mentor. His course on *Intermediate Machine Learning* was my first deep dive into neural networks, and his insight, patience, and steady encouragement later guided every stage of this thesis. From shaping the initial research question to refining the final manuscript, he treated each of my half-formed ideas with seriousness, asking the hard questions that ultimately made the work stronger. I could not have asked for a more generous mentor; under his tutelage I leave Yale a more confident machine-learning researcher.

I am equally indebted to Awni Altabaa, whose enthusiasm for all things oscillatory and symbolic turned countless white-board sketches into clear experiments. Whether debugging code at odd hours or challenging me to justify an architectural decision, Awni's feedback sharpened both my thinking and my craft.

My heartfelt thanks go to my closest friends, Noah Silverberg and Manuel Perez. Noah's grounding in physics and Manuel's perspective in neuroscience supplied precisely the cross-disciplinary intuition this project needed—often leaving us debating for hours well past midnight on this project's direction. Their blend of intellectual rigor and good humor has been the emotional center of my undergraduate years; every breakthrough in these pages echoes our shared laughter.

Finally, I am a reflection of everyone whose path I have crossed. To my professors, friends, and family—those who offered advice, asked a difficult question, or simply believed I could do the work—thank you. Whatever successes appear in this thesis are as much yours as mine.

1 Introduction

1.1 Relational Reasoning and the Binding Problem

Human intelligence exhibits a exceptional capacity for relational reasoning – the ability to identify abstract relationships (e.g., same vs. different, analogies) independent of the specific entities involved. This capacity allows humans to generalize knowledge across contexts with remarkable efficiency, often requiring only a handful of examples to extract abstract rules that can then be systematically applied to new scenarios (Gentner 1983). Achieving such flexibility in artificial neural networks remains challenging, in large part due to the binding problem: *how can neural representations dynamically bind features into structured relational units?*

In the brain, one hypothesized solution is binding by synchrony (Singer 2007). Early neural models suggested that neurons representing features of the same perceptual object could synchronize their firing, thereby implicitly "tagging" those features as belonging together (von der Malsburg and Schneider 1986). Subsequent neurophysiological work provided evidence synchronous oscillations may define relational structures across neural signals (Singer 1999). Existing cognitive models, such as LISA, have implemented synchrony-based binding to represent structured relations within neurally plausible frameworks (Hummel and Holyoak 2003). These insights motivate the exploration of biologically-plausible binding mechanisms in modern deep learning architectures for relational reasoning.

1.2 Symbolic vs. Neural Approaches to Relational Reasoning

Traditional approaches to relational reasoning in AI have often relied on symbolic representations or explicit logical structures. Symbolic systems can naturally represent abstract relations and achieve systematic generalization by treating relations as variables or rules (Lake et al. 2017). While symbolic methods have shown strong generalization from few examples by leveraging compositional abstractions, they typically require pre-defined rules or exhaustive search, which becomes intractable when seeking to emulate human-like reasoning (Lake et al. 2017).

Pure deep learning approaches learn directly from data and have achieved remarkable successes in pattern recognition; yet traditional neural networks struggle with data-efficient learning of abstract relations (Barrett et al. 2018). Networks without the right inductive biases often overfit to superficial features rather than truly abstracting the relational structure, leading to failures on out-of-distribution relational tasks. This tension between the data inefficiency of purely neural methods and the inflexibility of symbolic methods has prompted interest in approaches that can give neural networks a more symbol-like capacity for relational abstraction.

1.3 The Relational Bottleneck Principle

One promising idea emerging from recent literature is the "relational bottleneck" principle (Webb et al. 2024). The relational bottleneck is an architectural inductive bias that forces a learning system to focus on relations while suppressing object-specific information. In practice, this means constraining the information flow such that only abstract relational structure (and not low-level feature details) is passed forward for further processing. By design, this encourages the network to form an internal representation of relationships (such as "X and Y are the same") divorced from the particular attributes of X or Y.

Several neural architectures have implemented this idea and demonstrated gains in data efficiency and generalization on relational reasoning tasks. For instance, the Emergent Symbol Binding Network (ESBN) uses a dual memory LSTM to separate "sensory" vs. "relational" code, enabling rapid learning of abstract rules from few examples and near-perfect transfer to novel entities (Webb et al. 2021). More recently, researchers incorporated an explicit relational module (the Abstractor) into a Transformer, which uses a form of relational cross-attention to enforce a relational bottleneck at each layer (Altabaa et al. 2024).

These approaches illustrate how introducing architectural biases for relational pro-

cessing – effectively giving the network a dedicated channel or mechanism to encode "relations" apart from raw features – can bridge the gap between symbolic and deep learning methods. Such networks do not require hand-coded symbols a priori, yet they learn abstract relational representations that behave analogously to symbols (e.g., variables that can be applied to new instances).

1.4 Neural Oscillations for Binding

While a relational bottleneck can be engineered in various ways, an intriguing question is whether biologically-inspired dynamics could naturally give rise to such a bottleneck. Cognitive neuroscience increasingly supports the idea that oscillatory dynamics naturally facilitate relational abstraction and structured memory. Specifically, brain oscillations across frequency bands can temporally segment and group information, creating dynamic "slots" for encoding sequential or relational data (Lisman and Jensen 2013).

Theta-gamma oscillations in the hippocampus illustrate this concept: gamma cycles encode individual items, while their sequential ordering is determined by their respective phases within the theta rhythm. Empirical evidence shows that this phase-based encoding directly predicts sequence memory performance (Heusser et al. 2016). Such synchronization and phase-locking exemplify emergent binding mechanisms, as temporally coordinated neural activity clusters related items.

In relational reasoning contexts, synchrony naturally isolates abstract relations by synchronizing neurons encoding a specific relation (e.g., "relation R") while maintaining asynchronous firing with neurons representing other relations. Thus, synchrony functions as a carrier of relational information, clearly segregating different relational structures for downstream processing.

1.5 Kuramoto Oscillator Networks in Deep Learning

Building on these insights, recent approaches integrate Kuramoto oscillator networks into deep learning to enhance flexible binding and abstraction. The Kuramoto model mathematically describes coupled phase oscillators capable of spontaneous synchronization (Kuramoto 1975).

Specifically, Artificial Kuramoto Oscillatory Neurons (AKOrNs) have been proposed as a differentiable alternative to conventional activation units in neural networks (Miyato et al. 2025), where each neuron's internal phase evolves over time according to Kuramoto dynamics. Neurons in AKOrN-based layers can phase-lock (synchronize) or de-synchronize with each other based on their coupling, naturally forming coherent groups or "assemblies."

This synchrony effectively creates a self-organized clustering mechanism, allowing neurons to collectively encode abstract concepts or objects. Miyato et al. (2025) showed that inserting Kuramoto oscillatory neurons into deep networks enabled unsupervised object discovery in images, and enhanced generalization on tasks requiring reasoning under distribution shifts. Notably, these benefits emerge from dynamical interactions (synchrony/competition) between oscillatory units rather than explicit symbol manipulation, thus encouraging the formation of a relational bottleneck that filters noise and retains only high-level relational information.

1.6 Our Contribution

We introduce a novel neural architecture using Kuramoto-based oscillator synchrony for relational reasoning. Specifically, we implement a network of coupled phase oscillators and train it on two paradigmatic relational tasks:

1. A basic same/different classification task, in which the model must decide whether a pair of stimuli are identical or not.
2. A Relational Match-to-Sample (RMTS) task, in which the model must infer an abstract relation from a sample pair (e.g., "same") and then identify which of two target pairs exhibits that same relation.

These tasks require the network to form an abstract concept ("same" or "different") that applies across different items and to apply it in new contexts – testing relational generalization. We show that our oscillator-based model can learn to solve the same/different task efficiently by forming synchronized assemblies corresponding to "same" vs. "different" relations.

Crucially, the learned relational representation is transferable: after training on same/different discrimination, the model can be adapted to the RMTS task with minimal additional training, successfully carrying over its notion of "sameness" to new stimuli and task structure—providing evidence that oscillatory dynamics can enable flexible abstractions.

Our primary contribution is demonstrating that differentiable, Kuramoto-based neural dynamics provide biologically plausible relational abstraction and systematic generalization across relational reasoning tasks.

2 Approach

Our approach leverages oscillator dynamics inspired by neural synchrony to address relational reasoning tasks. Unlike traditional neural networks that rely primarily on static representations, we employ binding-by-synchrony—a biologically-motivated method for dynamically grouping features into relational structures. This section describes our strategy.

2.1 Oscillatory Dynamics as a Binding Mechanism

The binding problem – how neural systems integrate distributed features into coherent representations – is fundamental in both neuroscience and artificial intelligence. One promising neurobiological solution is binding-by-synchrony, the idea that neurons synchronize their firing to group distributed representations without explicit structural connections.

We implement binding-by-synchrony through a network of Kuramoto oscillators. In our approach, each visual feature is represented by an oscillator evolving on a unit sphere. Rather than relying solely on feature similarity for binding, we allow these oscillators to interact through learned coupling matrices, enabling them to synchronize based on both feature similarity and relational structure and thereby naturally encoding abstract relationships such as "sameness."

For instance, for two identical objects, their corresponding feature oscillators will strongly synchronize due to the between-object coupling, reflecting their underlying similarity. Conversely, different objects will produce distinct synchronization patterns where only certain feature dimensions align, capturing the relationship between dissimilar stimuli. These synchronization patterns serve as "signatures" of abstract relations that can generalize across different inputs.

2.2 Coherence as a Relational Bottleneck

Our approach enforces a "relational bottleneck" by measuring coherence—synchronization across oscillators—to abstract relationships away from specific input features. After running oscillator dynamics, we compute a coherence vector capturing relational information while discarding absolute feature details and stimulus-specific information. This bottleneck ensures downstream classifiers rely *solely* on relational structure, effectively enforcing abstraction.

This mechanism is conceptually similar to the approach implemented in the ESBN (Webb et al. 2021), which achieves a relational bottleneck through its external memory architecture. However, our oscillatory approach implements this bottleneck through dynamics rather than through explicit architectural constraints.

2.3 Two-Stage Transfer Learning Framework

We implement a hierarchical approach to solving the two relational reasoning tasks:

Stage 1 (Same/Different Learning): First, we train our model end-to-end to solve the same/different discrimination task. This model learns to map pairs of images to a coherence

vector that captures their relational structure, and then classifies this vector as representing "same" or "different" relations.

Stage 2 (RMTS Transfer Learning): After the same/different model has been trained, we freeze its parameters and use it as a feature extractor for the RMTS task. The pre-trained model processes each image pair (source and two targets) to generate coherence vectors, which are then concatenated and fed into a new classifier trained to identify which target pair matches the relation of the source pair.

This approach tests if coherence vectors genuinely capture transferable abstract relations. Success in transferring relational knowledge from simpler to more complex tasks (like the higher-order relational reasoning required by RMTS) demonstrates that the oscillatory binding mechanism encodes generalizable relational concepts, rather than merely memorizing specific stimuli.

2.4 Comparison with Alternative Binding Approaches

Our oscillatory binding approach differs fundamentally from other solutions to the binding problem implemented in neural architectures:

Attention-Based Binding: Methods like Transformers use static attention weights to link entities within single processing steps. In contrast, our Kuramoto-based approach employs temporal dynamics that allow binding patterns to emerge naturally from interactions between oscillators, potentially capturing more complex relational structures.

Memory-Based Binding: The ESBN model implements binding through an external memory with a key-value structure, enabling indirect references between abstract variables and concrete values. While this approach has proven effective for relational reasoning, it requires an explicit architectural separation between relational and sensory processing. Our approach achieves similar functional separation through oscillatory dynamics rather than through explicit memory structures.

Tensor Product Binding: Classical tensor product approaches explicitly bind roles and fillers through outer products. While powerful, these approaches often suffer from dimensionality explosion as the number of roles and fillers increases. Our method achieves similar functional separation through emergent oscillatory dynamics.

Unlike these alternatives, our approach is directly inspired by theories of neural binding in biological systems. The synchronization of oscillator populations provides a biologically plausible mechanism for flexible binding that can be fully integrated into differentiable neural architectures. This alignment with biological principles may contribute to more human-like patterns of abstraction and generalization in relational reasoning tasks.

3 Model Architecture

Our model architecture integrates oscillatory dynamics into an end-to-end differentiable neural network framework. Inspired by the generalized Kuramoto model Miyato et al. (2025), we leverage the natural binding properties of synchronized oscillators to create a relational reasoning system. The architecture consists of four main components: (1) a feature extraction module, (2) a Kuramoto oscillator network, (3) a coherence measurement module, and (4) task-specific classification components. In this section, we detail each component and how they interact to form a complete relational reasoning system.

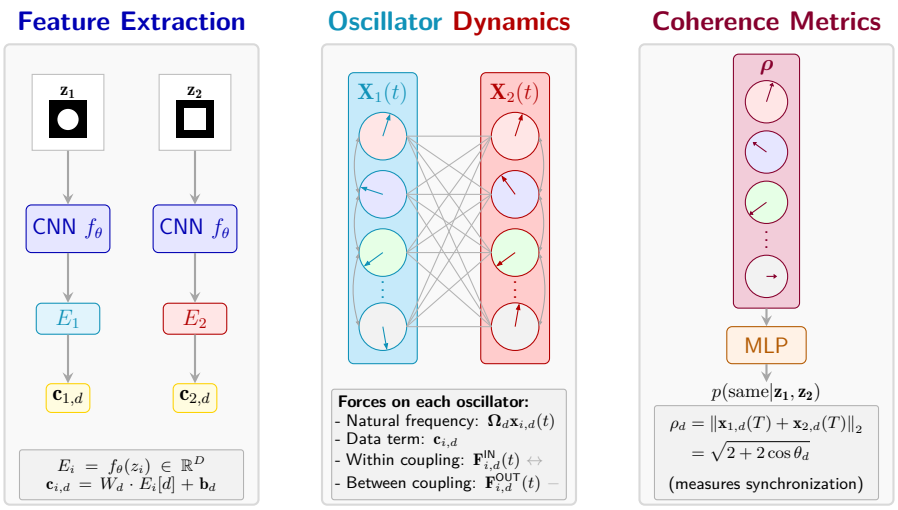


Figure 1: Schematic overview of our oscillatory binding architecture for relational reasoning. **Left:** Input stimuli (z_1, z_2) are processed by a convolutional neural network into feature embeddings $(E_1, E_2 \in \mathbb{R}^D)$, which drive oscillator dynamics via conditional stimuli $(\mathbf{c}_{i,d})$. **Middle:** Oscillators $(\mathbf{x}_{i,d})$ evolve on the unit circle $(N = 2)$ driven by natural frequencies, data inputs, and learned coupling matrices—both within $(\mathbf{J}^{\text{IN}})$ and between objects $(\mathbf{J}^{\text{OUT}})$. **Right:** A coherence vector $(\boldsymbol{\rho})$ measures synchronization between corresponding oscillators, forming a relational bottleneck. This relational representation is classified by a simple MLP to determine whether inputs exhibit the same or different relations.

3.1 Notation

We first define notation for clarity:

- $i, j \in \{1, 2, ...\}$: Indices denoting different objects or inputs
- $d, d' \in \{1, 2, ..., D\}$: Indices denoting feature dimensions
- $t \in \{0, 1, ..., T\}$: Time step in the Kuramoto dynamics
- $D \in \mathbb{N}$: Number of feature dimensions (and oscillators) per object
- $N \in \mathbb{N}$: Dimensionality of each oscillator (rotating dimensions)
- $T \in \mathbb{N}$: Total number of time steps for Kuramoto dynamics
- z_i: Input image for object i
- $E_i \in \mathbb{R}^D$: Embedding vector of object i extracted by CNN
- $\mathbf{x}_{i,d}(t) \in \mathbb{R}^N$: State of oscillator for object i, feature d at time t
- $\mathbf{X}_i(t) \in \mathbb{R}^{D \times N}$: Complete oscillator state for object i at time t
- $\mathbf{c}_{i,d} \in \mathbb{R}^N$: Conditional stimulus for oscillator $\mathbf{x}_{i,d}$
- $\mathbf{J}^{\text{IN}}_{d,d'} \in \mathbb{R}^{N \times N}$: Within-object coupling matrix between features d and d'
- $\mathbf{J}^{\text{OUT}}_{d,d'} \in \mathbb{R}^{N \times N}$: Between-object coupling matrix between features d and d'
- $\mathbf{\Omega}_d \in \mathbb{R}^{N \times N}$: Natural frequency matrix for feature dimension d
- $\rho_d \in \mathbb{R}$: Coherence measure for feature dimension d
- $\boldsymbol{\rho} \in \mathbb{R}^D$: Coherence vector across all feature dimensions

3.2 Feature Extraction

The feature extraction module translates raw pixel inputs into feature embeddings that drive the oscillator network. We deliberately use a simple convolutional architecture to focus our evaluation on the oscillatory binding mechanism rather than sophisticated feature extraction. The network consists of a two-layer convolutional neural network followed by a fully connected layer.

The CNN architecture progressively reduces spatial dimensions while extracting hier-

archical visual features, ultimately producing a compact embedding $E_i \in \mathbb{R}^D$ for each input image. Each dimension of this embedding corresponds to a visual feature that will drive a single oscillator in the next stage. This feature extractor is trained end-to-end with the rest of the model, allowing it to learn features specifically useful for the oscillatory dynamics that follow.

3.3 Kuramoto Oscillator Network

The core of our architecture is the Kuramoto oscillator network, which implements binding through synchronization dynamics. Following the vector-valued Kuramoto model formulation Miyato et al. (2025), we represent each feature dimension with an oscillator, enabling representations that facilitate relational reasoning.

3.3.1 Oscillator Representation

Each oscillator $\mathbf{x}_{i,d}(t)$ is represented as a point on the N-dimensional unit sphere: $\mathbf{x}_{i,d}(t) \in \mathbb{R}^N$ with $\|\mathbf{x}_{i,d}(t)\|_2 = 1$. Each oscillator is initialized from its corresponding embedding feature through a learned transformation:

$$\mathbf{c}_{i,d} = W_d \cdot E_i[d] + \mathbf{b}_d \in \mathbb{R}^N$$

$$\mathbf{x}_{i,d}(0) = \frac{\alpha \cdot \mathbf{c}_{i,d} + \eta_{i,d}}{\|\alpha \cdot \mathbf{c}_{i,d} + \eta_{i,d}\|_2}$$

where $W_d \in \mathbb{R}^N$ and $\mathbf{b}_d \in \mathbb{R}^N$ are learned parameters, α is a scaling factor, and $\eta_{i,d} \sim \mathcal{N}(0, \sigma^2)$ is small random noise. This initialization serves two purposes: (1) it aligns each oscillator with a transformed version of its feature value, providing a data-dependent starting point for dynamics, and (2) it introduces small perturbations to break perfect symmetry, which helps avoid degenerate solutions during optimization. The normalization ensures that each oscillator lies on the unit sphere.

3.3.2 Kuramoto Dynamics

The oscillators evolve according to Kuramoto dynamics, governed by the vector-valued differential equation from Miyato et al. (2025):

$$\dot{\mathbf{x}}_{i,d} = \mathbf{\Omega}_d \mathbf{x}_{i,d} + \text{Proj}_{\mathbf{x}_{i,d}}(\mathbf{c}_{i,d} + \mathbf{F}^{\text{IN}}_{i,d} + \mathbf{F}^{\text{OUT}}_{i,d})$$

where $\text{Proj}_{\mathbf{x}}(\mathbf{y}) = \mathbf{y} - (\mathbf{x} \cdot \mathbf{y})\mathbf{x}$ is the projection operator that ensures that the oscillator state remains constrained to the unit sphere.

Three key forces govern oscillator movement:

Natural Frequency Term: Each oscillator intrinsically rotates at a frequency determined by a skew-symmetric matrix $\mathbf{\Omega}_d \in \mathbb{R}^{N \times N}$ that generates rotations on the unit sphere:

$$\mathbf{F}^{\text{freq}}_{i,d}(t) = \mathbf{\Omega}_d \mathbf{x}_{i,d}(t)$$

These matrices are sampled from a von Mises distribution. The skew-symmetry of $\mathbf{\Omega}_d$ ensures that $\mathbf{x}^T_{i,d}\mathbf{\Omega}_d\mathbf{x}_{i,d} = 0$, preserving the unit norm constraint. This frequency term provides oscillator diversity, preventing trivial synchronization states and thus acting as an implicit regularization mechanism that forces stronger coupling to overcome the natural frequencies for synchronization to occur.

Within-Object Coupling: This force drives synchronization between oscillators representing different features of the same object:

$$\mathbf{F}^{\text{IN}}_{i,d}(t) = \sum_{d'=1}^{D} \mathbf{J}^{\text{IN}}_{d,d'} \mathbf{x}_{i,d'}(t)$$

where $\mathbf{J}^{\text{IN}}_{d,d'} \in \mathbb{R}^{N \times N}$ defines coupling strength between features d and d' within the same object. This coupling implements a form of competitive learning through binding, where correlated features within an object tend to synchronize their representations.

Between-Object Coupling: This force drives synchronization between corresponding features across both objects:

$$\mathbf{F}_{i,d}^{\text{OUT}}(t) = \sum_{d'=1}^{D} \mathbf{J}_{d,d'}^{\text{OUT}} \mathbf{x}_{3-i,d'}(t)$$

where $\mathbf{J}_{d,d'}^{\text{OUT}} \in \mathbb{R}^{N \times N}$ defines coupling strength between features across objects. This coupling is critical for relational reasoning, as it allows the model to learn which features should be compared across objects when determining their relationship.

The coupling matrices use Xavier uniform initialization to achieve appropriate variance at initialization.

3.3.3 Update Procedure

To discretize the continuous Kuramoto dynamics, we use a first-order Euler integration scheme to enforce the unit-sphere constraint. Each oscillator is updated at timestep t as follows:

$$\begin{aligned}
\mathbf{F}_{i,d}^{\text{total}}(t) &= \mathbf{c}_{i,d} + \mathbf{F}_{i,d}^{\text{IN}}(t) + \mathbf{F}_{i,d}^{\text{OUT}}(t) \\
\Delta\mathbf{x}_{i,d}(t) &= \mathbf{F}_{i,d}^{\text{freq}}(t) + \text{Proj}_{\mathbf{x}_{i,d}(t)}\left(\mathbf{F}_{i,d}^{\text{total}}(t)\right) \\
\mathbf{x}_{i,d}(t+1) &= \frac{\mathbf{x}_{i,d}(t) + \gamma \cdot \Delta\mathbf{x}_{i,d}(t)}{\|\mathbf{x}_{i,d}(t) + \gamma \cdot \Delta\mathbf{x}_{i,d}(t)\|_2}
\end{aligned}$$

where γ is the step size controlling the numerical stability of integration. The update consists of:

1. Computing the total force from conditional stimuli and coupling interactions
2. Projecting this force onto the tangent space to preserve the spherical constraint
3. Adding the natural frequency term (which is already tangent to the sphere)
4. Applying the step size and renormalizing to correct for any numerical drift

Iterating this procedure for T timesteps allows the oscillator states to converge into stable synchronization patterns that encode the relational structure between inputs.

3.4 Energy Function

Under specific symmetric coupling conditions, Kuramoto oscillator networks minimize the following energy function (Aoyagi 1995; Miyato et al. 2025; Wang and Roychowdhury 2017). In particular, when the coupling matrices $\mathbf{J}_{ij}$ satisfy symmetry conditions such as $\mathbf{J}_{ij} = J_{ij}\mathbf{I}$, with J_{ij} symmetric and real-valued, and the frequency matrices $\mathbf{\Omega}_i$ are uniform ($\mathbf{\Omega}_i = \mathbf{\Omega}$) with the constraint $\mathbf{\Omega}\mathbf{c}_i = \mathbf{0}$, the network dynamics minimize:

$$E = -\frac{1}{2}\sum_{i,j} \mathbf{x}_i^\top \mathbf{J}_{ij} \mathbf{x}_j - \sum_i \mathbf{c}_i^\top \mathbf{x}_i$$

This energy consists of two intuitive terms. The first, coupling energy, encourages synchronization among oscillators according to the coupling strengths, thereby promoting coherent representations. The second term, stimulus alignment, encourages oscillators to closely align with their data-driven inputs, preventing trivial or uniform synchronization states. However, in general conditions without these symmetric constraints, the oscillator network dynamics do not strictly guarantee energy minimization, although empirical results often still show stable decreases in energy and improved task performance.

In our two-object architecture, we explicitly decompose this energy as:

$$E_{\text{total}} = E_{\text{coupling}} + E_{\text{stimulus}} \tag{1}$$

with

$$E_{\text{coupling}} = -\frac{1}{2}\left(\mathbf{x}_1^\top \mathbf{J}^{\text{IN}} \mathbf{x}_1 + \mathbf{x}_2^\top \mathbf{J}^{\text{IN}} \mathbf{x}_2 + \mathbf{x}_1^\top \mathbf{J}^{\text{OUT}} \mathbf{x}_2 + \mathbf{x}_2^\top \mathbf{J}^{\text{OUT}} \mathbf{x}_1\right),$$

and

$$E_{\text{stimulus}} = -\left(\mathbf{c}_1^\top \mathbf{x}_1 + \mathbf{c}_2^\top \mathbf{x}_2\right).$$

Lower energy states correspond intuitively to synchronized oscillator configurations, reflecting stronger relational bindings.

3.5 Coherence Measurement

After the Kuramoto dynamics evolve over T time steps, we measure coherence to quantify synchronization between corresponding oscillators, capturing the relational structure between inputs. For each feature dimension d, coherence is computed as:

$$\rho_d = \|\mathbf{x}_{1,d}(T) + \mathbf{x}_{2,d}(T)\|_2 = \sqrt{2 + 2\cos\theta_d}$$

where θ_d denotes the angle between oscillators $\mathbf{x}_{1,d}(T)$ and $\mathbf{x}_{2,d}(T)$. Coherence ranges from $\rho_d = 2$ (perfect alignment, $\theta_d = 0$) to $\rho_d = 0$ (complete opposition, $\theta_d = \pi$), providing a continuous measure of similarity.

The resulting coherence vector $\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_D)$ has various properties advantageous for relational reasoning: (1) it is invariant to global rotations, focusing solely on relative oscillator alignment; (2) it compresses oscillator states into a fixed-size relational representation independent of oscillator dimensionality; and (3) it naturally encodes relationships through synchronization patterns, arising directly from oscillator dynamics.

Thus, coherence functions as a relational bottleneck, distilling inputs into abstract relational representations free from absolute stimulus information. This architectural choice promotes generalization by constraining downstream classifiers to reason based solely on relational structure.

3.6 Classification Components

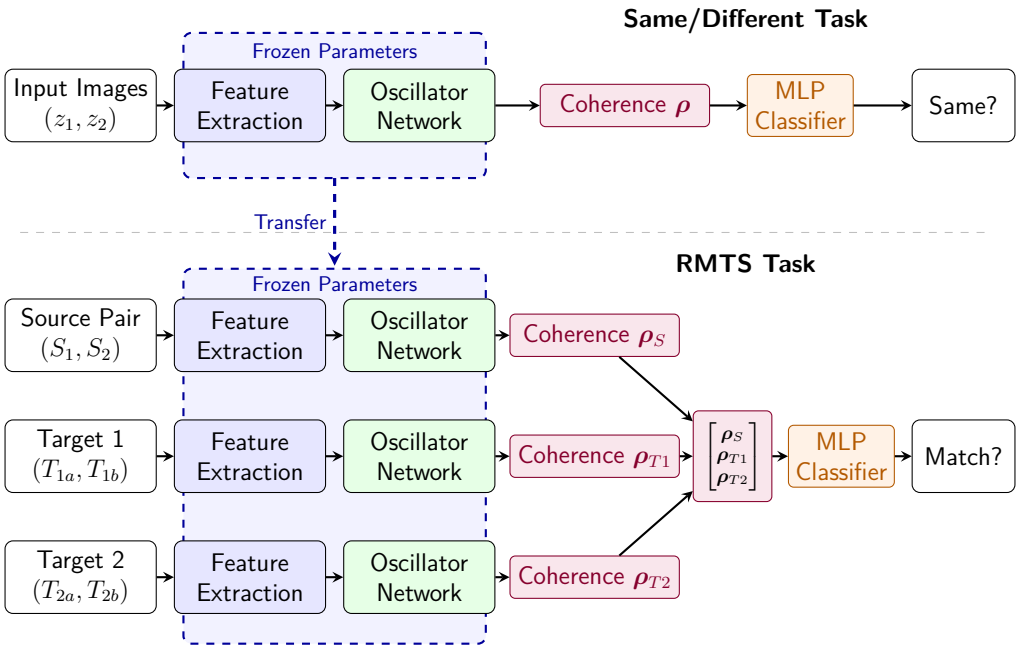


Figure 2: Transfer learning architecture. **Top:** On the Same/Different task, input image pairs pass through feature extraction and oscillator networks to produce a coherence vector, classified by an MLP. **Bottom:** For the RMTS task, coherence vectors from the source and target pairs—generated using parameters frozen from Same/Different training—are concatenated and classified by a new RMTS-specific MLP.

3.6.1 Same/Different Classification

For the same/different classification task, the coherence vector serves as input to a simple multilayer perceptron (MLP). This classifier consists of one hidden layer with ReLU activation followed by a sigmoid output unit, yielding a probability that the input pair represents the "same" relation. We deliberately employ a minimal classifier to emphasize the representational power of the oscillator network: synchronization dynamics alone should encode the

necessary relational information, leaving the MLP to perform only straightforward relational decoding.

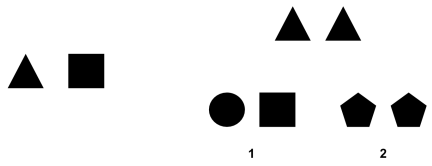
3.6.2 RMTS Classification

For the relational match-to-sample (RMTS) task, we first extract coherence vectors from each of the three input pairs (source, target 1, and target 2). These three coherence vectors are concatenated and fed into a new MLP classifier designed specifically for RMTS. The MLP consists of a single hidden layer with ReLU activation, followed by a softmax output layer that predicts which target pair shares the same relational structure as the source pair.

Critically, we perform the transfer learning approach described in Section 2.3: after training the oscillator network and coherence measurement on the simpler same/different task, we freeze these parameters and reuse them directly for RMTS. Thus, the RMTS classifier tests whether the coherence vectors genuinely encode abstract relational information transferable across tasks. Successful transfer demonstrates that synchronization dynamics can generate generalizable relational representations suitable for higher-order reasoning.

4 Tasks

As mentioned, we evaluate our oscillatory binding approach on two relational reasoning tasks: same/different discrimination and relational match-to-sample (RMTS). These tasks were selected because they require abstract relational reasoning at increasing levels of complexity, rather than mere feature matching or pattern recognition. Both tasks use the same visual stimuli, allowing direct performance comparisons.



(a) Same/different (b) RMTS

Figure 3: Abstract rule learning tasks. (a) Same/different discrimination task: determine if two images are identical. (b) Relational match-to-sample (RMTS) task: determine which of two target pairs exhibits the same relation as the source pair.

4.1 Visual Stimuli

For all tasks, we employ the same set of 100 images used by Webb et al. (2021), in which each image represents a distinct Unicode character. These simple icons minimize irrelevant complexity, enabling a focused evaluation of relational reasoning capabilities. The images, presented as centered 32×32 grayscale characters, undergo random transformations (scale and positioning) during training to prevent pixel-level memorization and encourage abstract relation recognition.

4.2 Same/Different

The same/different discrimination task is arguably the most fundamental relational reasoning challenge. This task requires the model to classify whether two presented images depict the same character. This task, although conceptually simple for humans, poses significant challenges for neural networks, which often rely on extensive training to generalize the abstract relation of sameness to novel stimuli (Kim et al. 2018).
Formally, given a pair of images (x_1, x_2), the task is to output a binary classification:

$$y = \begin{cases} 1 & \text{if } x_1 \text{ and } x_2 \text{ show the same character} \\ 0 & \text{if } x_1 \text{ and } x_2 \text{ show different characters} \end{cases}$$

In our implementation, we construct balanced datasets with equal numbers of "same" and "different" pairs. This balance ensures the model cannot succeed by defaulting to a majority class.

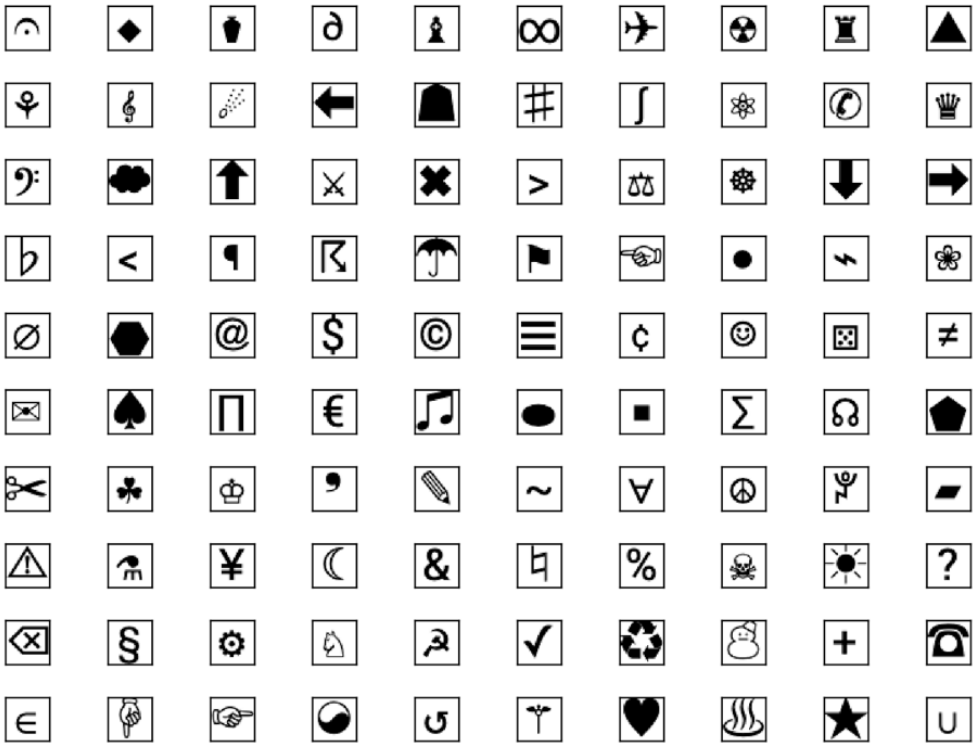


Figure 4: The 100 Unicode characters used in our experiments. During generalization tests, subsets are withheld from training and reserved for evaluation.

4.3 Relational Match-to-Sample (RMTS)

The relational match-to-sample task represents a higher-order extension of the same/different concept, requiring the model to reason about "relations between relations." In this task, the model is first presented with a source pair of images that establishes a relation (either "same" or "different"). It must then compare this relation to two target pairs, identifying which target pair exhibits the same relation as the source. Critically, the actual characters used in the source and target pairs are distinct, forcing the model to abstract the relationship rather than matching visual features directly.

This task is well-known in cognitive science research due to its reliance on advanced abstraction capabilities, distinguishing human reasoning from non-human animal performance (Premack 1983).

Formally, the task consists of a source pair (S_1, S_2) and two target pairs (T_{1a}, T_{1b}) and (T_{2a}, T_{2b}). Let $R(A, B)$ represent the relation between images A and B, where:

$$R(A,B) = \begin{cases} \text{“same”} & \text{if } A \text{ and } B \text{ show the same character} \\ \text{“different”} & \text{if } A \text{ and } B \text{ show different characters} \end{cases}$$

The model must determine which target pair has the same relation as the source pair:

$$y = \begin{cases} 0 & \text{if } R(S_1,S_2) = R(T_{1a},T_{1b}) \\ 1 & \text{if } R(S_1,S_2) = R(T_{2a},T_{2b}) \end{cases}$$

In our datasets, we construct balanced problems where each relation type ("same" and "different") appears equally often as the correct answer. As with the same/different discrimination task, this prevents the model from relying on trivial performance strategies.

4.4 Generalization Regimes

A key challenge in relational reasoning is generalizing abstract rules to novel stimuli. To systematically evaluate this capability, we consider three generalization regimes, each defined by the number of unique characters, m, presented during training. This approach modifies the setup of Webb et al. (2021), where m originally referred to characters withheld during training. Here, larger m corresponds to broader exposure during training and consequently less demanding generalization:

Extensive Training ($m = 95$): In this regime, 95 characters are used during training, with the remaining 5 reserved exclusively for testing. The model must apply learned relational concepts to a small set of novel (out-of-distribution) characters while leveraging familiarity with the overall character distribution.

Moderate Generalization ($m = 50$): Only half (50) of the characters appear during training, and the model must generalize to the remaining 50 novel characters at test time.

Limited Training ($m = 15$): With only 15 characters available in training, models must exhibit robust abstraction capabilities to generalize to 85 unseen characters at test time.

In each regime, training datasets contain 4,000 examples for $m = 95$, and 2,000 examples each for $m = 50$ and $m = 15$. Test sets consistently comprise 10,000 examples across all regimes. Each character is randomly resized (between 70% and 100%) and repositioned within the canvas, generating visual variability and preventing simple memorization and assessing the model's abstract relational reasoning across varying degrees of distributional shift. All models are trained for 50 epochs, ensuring consistent and fair comparisons across the various generalization scenarios.

Table 1: Training and test set sizes for each generalization regime.

Regime	Training Chars	Testing Chars	Train Size	Test Size	Epochs
Full	95	5	4,000	10,000	50
Moderate	50	50	2,000	10,000	50
Limited	15	85	2,000	10,000	50

5 Training Details

In this section, we describe our training methodology, including hyperparameters, optimization techniques, and evaluation procedures for both the Same/Different and RMTS tasks.

5.1 Model Configuration

All experiments utilized the oscillator-based model with consistent architectural parameters across tasks. The feature extractor consisted of a CNN with two convolutional layers (16 and 32 filters respectively, both using 5×5 kernels, stride 2, and padding 2), followed by a fully connected layer projecting to a 64-dimensional embedding space. For the oscillator network, we set embedding dimension $D = 64$ and oscillator dimension $N = 2$, allowing direct visualization of oscillator trajectories on the unit circle. We empirically found that 25 Kuramoto time steps ($T = 25$) with step size $\gamma = 0.1$ provided stable dynamics while maintaining computational efficiency.

The oscillator network used natural frequencies ($\mathbf{\Omega}_d$) sampled from a von Mises distribution with concentration parameter $\kappa = 2.0$, providing moderate rotational diversity without overwhelming the learned coupling. For the Same/Different task, we employed a single-hidden-layer MLP classifier with 64 hidden units, while the RMTS classifier used a similar architecture with an input dimension of $3D = 192$ (concatenated coherence vectors from three pairs).

5.2 Optimization

Both tasks were trained using the AdamW optimizer with a learning rate of 10^{-3} and weight decay of 10^{-4}. We employed a learning rate schedule consisting of a linear warmup

for the first 5% of training steps, followed by cosine annealing to a minimum rate of 10^{-6}. This schedule helped stabilize initial training while allowing for fine-grained optimization in later epochs.

To prevent gradient explosions in the oscillator dynamics, we applied gradient clipping with a maximum norm of 1.0. We used mixed-precision training (FP16) to accelerate computation without sacrificing model performance. All experiments were conducted using the PyTorch framework with the Accelerate library for distributed training capabilities.

5.3 Training Procedure

For the Same/Different task, we trained models for 50 epochs with a batch size of 64. The binary classification task was optimized using binary cross-entropy with logits (BCEWithLogitsLoss).

For the RMTS task, we employed the transfer learning approach described in Section 2.3. Specifically, after training on the Same/Different task, we:

1. Froze the parameters of the feature extractor and oscillator network
2. Generated coherence vectors for all source and target pairs
3. Trained a new MLP classifier on these coherence vectors for 50 epochs

The RMTS classifier was trained using standard cross-entropy loss (CrossEntropyLoss) to predict which target pair matched the relation in the source pair.

5.4 Data Processing

All images were preprocessed using a consistent transformation pipeline: random resizing (70-100% of original size) followed by random placement on a 32×32 canvas. This augmentation strategy prevented the model from relying on absolute positioning and encouraged abstraction of the visual features.

For each generalization regime ($m \in \{15, 50, 95\}$), we constructed datasets as described in Section 4.4. Training sets contained $n = 2,000$ examples for the $m = 15$ and $m = 50$ conditions, and $n = 4,000$ examples for the $m = 95$ condition. Test sets consistently comprised 10,000 examples across all regimes. All datasets were balanced to contain equal numbers of "same" and "different" pairs for the Same/Different task, and balanced target selections for the RMTS task.

5.5 Alternative Architectures

To assess our oscillatory binding model, we evaluate its performance against several baseline and ablated architectures, designed to highlight the specific contributions of oscillator dynamics and binding mechanisms.

5.5.1 CNN+MLP Baseline

We first implement a straightforward baseline using a standard CNN combined with an MLP classifier, omitting any explicit binding or synchronization mechanisms. To ensure comparability, this model employs the same convolutional layers as our oscillator-based architecture. For the same/different task, the CNN processes each input image independently to produce feature embeddings, computes a coherence vector representing their relationship, and passes this vector through a two-layer MLP with ReLU activations before generating classification outputs. For the RMTS task, the CNN processes each image pair separately, and the resulting representations are concatenated and classified using an MLP.

5.5.2 Ablated Oscillator Models

We further investigate the contributions of individual components within our oscillatory binding network through targeted ablations:

No Between-Object Coupling ($\mathbf{J^{OUT}} = 0$): In this variant, we eliminate the coupling term that synchronizes oscillators across objects, restricting the network to within-object synchronization only. Evaluating this model measures the necessity of cross-object interactions for successful relational abstraction.

No Natural Frequency ($\mathbf{\Omega} = 0$): Here, we remove the intrinsic oscillator rotation component, effectively eliminating the inherent regularization. This ablation tests the importance of oscillator diversity and rotational dynamics for synchronization and relational learning.

Symmetric Coupling Matrices with No Natural Frequency ($\mathbf{J^{IN}}, \mathbf{J^{OUT}}$ symmetric, $\mathbf{\Omega} = 0$): In this model, we enforce symmetry on both within-object ($\mathbf{J}^{\text{IN}}$) and between-object ($\mathbf{J}^{\text{OUT}}$) coupling matrices and remove the intrinsic natural frequency component ($\mathbf{\Omega} = 0$). This configuration theoretically ensures a monotonically decreasing energy landscape during oscillator synchronization, despite us not explicitly optimizing the energy function.

6 Results and Discussion

We now turn to an empirical assessment of oscillatory binding. We first examine in-distribution learning dynamics, determining whether synchrony accelerates the acquisition of relational structure. We then evaluate out-of-distribution generalization under the three data regimes introduced in Section 4.4. Unless stated otherwise, numbers are averaged over five random seeds and reported as mean ± standard error of the mean (SEM).

Finally, we investigate the internal mechanics of oscillator synchronization through energy landscapes and dynamic state trajectories, providing insights into the binding mechanisms underlying relational abstraction.

6.1 Training (In-Distribution) Performance and Convergence

We first present the in-distribution training accuracy results in Table 2, measured after 50 epochs (3,200 mini-batches), and examine their convergence trajectories in Figure 5 for the richest data regime ($m = 95$). All model configurations surpass 90% accuracy on both tasks, confirming their ability to effectively learn the training dataset. However, significant variations in the speed and trajectory of convergence underscore the functional importance of oscillatory binding mechanisms.

Table 2: Mean ± SEM training accuracy (%) after 50 epochs (3,200 training batches).

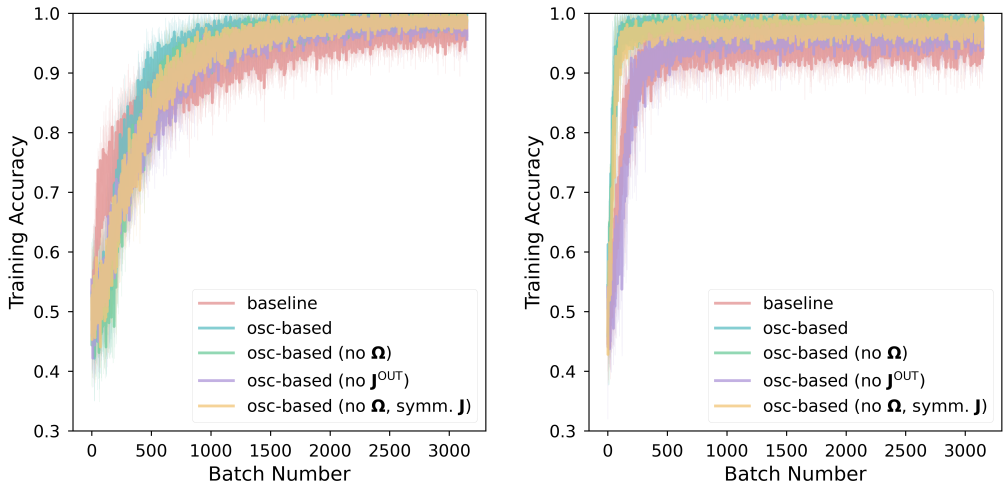
Model	**Same/Different Train Acc.**			**RMTS Train Acc.**		
	$m = 95$	$m = 50$	$m = 15$	$m = 95$	$m = 50$	$m = 15$
Baseline (CNN+MLP)	96.9 ± 0.2	95.6 ± 0.1	$\mathbf{99.7 \pm 0.1}$	94.7 ± 0.3	93.6 ± 0.4	$\mathbf{99.7 \pm 0.1}$
Oscillator (Default)	$\mathbf{99.2 \pm 0.1}$	$\mathbf{96.8 \pm 0.5}$	99.5 ± 0.2	$\mathbf{98.2 \pm 0.1}$	94.5 ± 0.8	99.3 ± 0.3
Oscillator (No $\mathbf{\Omega}$)	98.8 ± 0.3	96.1 ± 0.5	98.4 ± 0.5	97.9 ± 0.3	$\mathbf{94.5 \pm 0.9}$	97.6 ± 0.5
Oscillator (No $\mathbf{J}^{\text{OUT}}$)	98.2 ± 0.2	93.9 ± 1.0	97.5 ± 0.5	96.3 ± 0.3	90.5 ± 1.3	96.4 ± 0.7
Oscillator (Symm $\mathbf{J}$, No $\mathbf{\Omega}$)	98.8 ± 0.1	95.3 ± 0.5	98.5 ± 0.3	97.4 ± 0.3	92.6 ± 1.0	97.6 ± 0.4

Note: Best metrics **bolded**.

The default oscillator model consistently achieves the highest or near-highest training accuracies across all experimental conditions. This model is followed by the CNN+MLP baseline and the oscillator variant without intrinsic frequency differences ($\mathbf{\Omega}$). Notably, the oscillator variant without cross-object coupling ($\mathbf{J}^{\text{OUT}} = 0$) consistently exhibits a moderate decline in accuracy, indicating that cross-object coupling contributes to effective relational learning at the training stage.

These observations are visually reinforced by the accuracy curves shown in Figures 5a and 5b, highlighting that oscillator-based models equipped with $\mathbf{J}^{\text{OUT}}$ exhibit more rapid convergence compared to the baseline and no-between-coupling models. The accelerated learning dynamics support the hypothesis that learned cross-object coupling promotes relational binding through synchronization, thus enhancing learning speed.

In the RMTS task (Figure 5b), where feature extraction and oscillator parameters are fixed and only the final classifier layers are trained, oscillator-based models rapidly achieve over 95% accuracy within roughly 300 batches. This rapid adaptation demonstrates that coherence vectors generated by oscillator coupling inherently capture relational structure, allowing the MLP classifier to efficiently decode higher-order relational information. This aligns with the relational-bottleneck principle, wherein synchronization suppresses



(a) Same/different ($m = 95$) (b) RMTS ($m = 95$)

Figure 5: Training accuracy curves for each model variant across 50 epochs (3,200 training batches). Error bands represent standard error of the mean across 5 runs.

irrelevant sensory details and highlights abstract relational features.

In contrast, the CNN+MLP baseline and oscillator without cross-object coupling converge more slowly and plateau at approximately 95% accuracy. The slower convergence emphasizes the limited capacity of these models to spontaneously generate transferable relational representations. Although the CNN+MLP baseline, relying solely on the classifier for inter-object communication, achieves surprisingly high final accuracy, its slower convergence underscores the substantial computational burden placed on the classifier due to the absence of an explicit relational binding mechanism.

As earlier noted, ablating cross-object interactions seems to diminish the benefits derived from oscillatory binding, despite the model retaining within-object coupling and intrinsic frequency terms. The slowed convergence of the $\mathbf{J}^{\text{OUT}} = 0$ model highlights phase-locking across objects as a central mechanism facilitating rapid relational learning.

Conversely, removing intrinsic rotation matrices ($\mathbf{\Omega} = 0$) or imposing symmetrical coupling weights produces minimal effects on convergence rates, suggesting these elements primarily act as minor regularizing factors rather than essential structural components.

6.2 Generalization (Out-of-Distribution) Performance

Table 3: Mean ± SEM test accuracy (%) across tasks and data regimes.

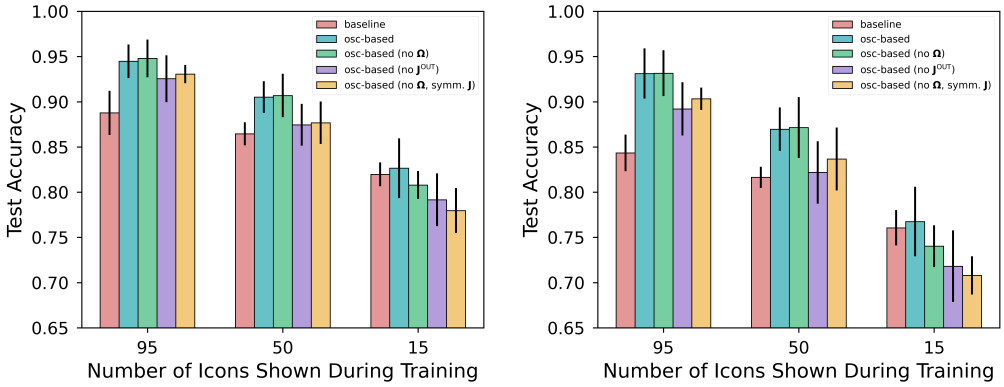
Model	**Same/Different Test Acc.**			**RMTS Test Acc.**		
	$m = 95$	$m = 50$	$m = 15$	$m = 95$	$m = 50$	$m = 15$
Baseline (CNN+MLP)	88.8 ± 1.1	86.5 ± 0.6	82.0 ± 0.6	84.3 ± 0.9	81.6 ± 0.5	76.1 ± 0.9
Oscillator (Default)	94.5 ± 0.8	90.5 ± 0.8	$\mathbf{82.7 \pm 1.5}$	93.1 ± 1.2	87.0 ± 1.1	$\mathbf{76.8 \pm 1.7}$
Oscillator (No $\mathbf{\Omega}$)	$\mathbf{94.8 \pm 0.9}$	$\mathbf{90.7 \pm 1.1}$	80.8 ± 0.7	$\mathbf{93.2 \pm 1.1}$	$\mathbf{87.2 \pm 1.5}$	74.0 ± 1.0
Oscillator (No $\mathbf{J}^{\text{OUT}}$)	92.5 ± 1.2	87.5 ± 1.0	79.2 ± 1.3	89.2 ± 1.3	82.2 ± 1.5	71.8 ± 1.8
Oscillator (Symm $\mathbf{J}$, No $\mathbf{\Omega}$)	93.1 ± 0.5	87.7 ± 1.0	78.0 ± 1.1	90.3 ± 0.6	83.7 ± 1.6	70.8 ± 0.9

Note: Best metrics **bolded**.

We next evaluate how well the trained models generalize to novel objects not seen during training. Table 3 reports test accuracies for each model on held-out sets for the Same/Different and RMTS tasks under the three generalization regimes.

Two trends are immediately apparent: First, the default oscillator model maintains

a consistent advantage over the purely feed-forward CNN + MLP baseline in every regime and on both tasks. The margin is largest when abundant training identities are available ($m = 95$), where the Same/Different accuracy rises from 88.8% to 94.5%, and the RMTS accuracy from 84.3% to 93.1%. Second, the benefit of oscillatory binding narrows—but does not disappear—as the amount of seen characters shrinks. When only fifteen characters are available for training, accuracies for the baseline and the full oscillator differ by less than a percentage point. This pattern suggests that cross-object synchrony does not merely promote rote memorization of low-level features; instead, it supplies inductive structure that is most useful when the learner has already encountered a modest diversity of tokens from which to abstract a relation.



(a) Same/different

(b) RMTS

Figure 6: Test accuracy across different generalization regimes. Error bars represent standard error of the mean across 5 runs.

The ablation results help to locate the source of this advantage. Removing the between-object coupling $\mathbf{J}^{\text{OUT}}$ uniformly depresses test accuracy, erasing much if not all of the oscillator's gain over the baseline. Because the CNN encoder and within-object couplings remain intact, the deterioration isolates the unique contribution of cross-object phase-locking. In other words, the same recurrent dynamics that accelerated training (Section 6.1) also furnish representations that transfer to unseen characters, corroborating our hypothesis that $\mathbf{J}^{\text{OUT}}$ is the principal catalyst for relational learning.

Eliminating the natural frequency term ($\boldsymbol{\Omega} = 0$), on the other hand, has a subtler effect. With plentiful data, the no-$\boldsymbol{\Omega}$ variant achieves the best numerical scores, but its advantage diminishes as the data regime becomes more challenging; by $m = 15$ it underperforms the default oscillator. Natural-frequency drift ($\boldsymbol{\Omega}$) seems to act as a data-dependent regularizer. With many training characters it injects superfluous phase noise, so removing it yields a small accuracy gain. When training characters are scarce, however, the drift may mitigate premature global synchrony, forcing the model to rely on learned cross-object couplings and thus improving its relational signal.

Moreover, imposing symmetric couplings while simultaneously zeroing the natural frequency results in dynamics that provably minimize the classical Kuramoto energy. However, the model does not match the performance of either the default or the no-$\boldsymbol{\Omega}$ variant. Evidently, monotonic energy descent appears to be insufficient for robust relational coding; on the contrary, mild asymmetries in the coupling weights and mild rotational drift appear to diversify the transient phase trajectories, leading to more effective relational representations that are easier for the classifier to discriminate.

6.3 Evolution of Energy Landscape

To better understand how relational abstraction emerges from oscillator synchronization, we examined the special case in which the within- and between-object coupling matrices are

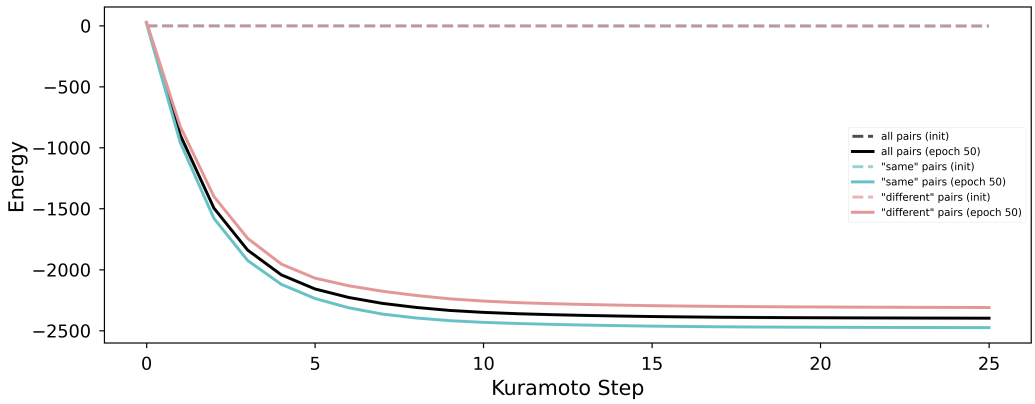


Figure 7: Energy evolution during Kuramoto dynamics for the oscillator model with symmetric coupling and no natural frequency. Energy values were computed using the first training minibatch, averaged across all pairs, "same" pairs only, and "different" pairs only.

symmetric and the intrinsic rotation term is ablated, ($\mathbf{\Omega} = 0$). Under these constraints the continuous-time vector-valued Kuramoto flow becomes a gradient descent on the Lyapunov function defined in Eq. 1; consequently the discrete Euler update used in the network should, up to numerical error, monotonically lower the energy at every iteration. Since no additional forces are present, the only route to a low-energy configuration is to align phases according to the learned couplings, so the trajectory of the energy directly reveals the emergence of synchrony-based bindings.

Figure 7 depicts the energy landscape before (initialization) and after training (epoch 50). Prior to optimization the couplings are random and the system expresses only weak collective behavior: energy remains effectively constant across the 25 simulation steps and the curves for "same'' and "different'' pairs are indistinguishable. The high initial energy indicates that oscillators start from largely de-synchronised states, and the absence of class separation confirms that the untrained network has yet to adapt to the relational structure of the stimuli and thus possesses no intrinsic bias toward the sameness relation.

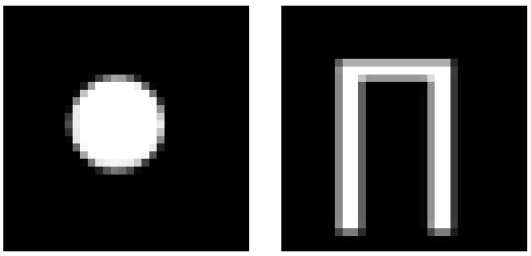
Training substantially transforms the dynamics. After training the model, the overall energy level decreases sharply, indicating that the couplings now impose much stronger attractive forces among related oscillators. Crucially, a pronounced separation appears: stimulus pairs that have "same'' relations reach markedly lower minima than those labeled "different.'' This divergence quantifies the relational information that the network has distilled into its dynamics. A low final energy corresponds to tight phase locking between related feature oscillators across the two objects, whereas a higher plateau indicates residual de-synchronization characteristic of mismatched inputs. The coherence vector extracted at the bottleneck therefore inherits a significant margin between the two relational categories, simplifying the downstream classifier's ability to discriminate relations and explaining the faster convergence and generally superior generalization reported earlier.

The fact that this class-specific energy gap emerges without any explicit constraint on the training objective affirms the central thesis of the paper: relational abstraction can materialize as an *emergent property* of oscillator synchronization. By demonstrating that the artificial network can spontaneously organizes its energy landscape so that "sameness'' corresponds to lower energy states than "difference,'' we provide concrete evidence for a physically grounded relational code—evidence that complements the performance metrics and ablation studies presented in Section 6.

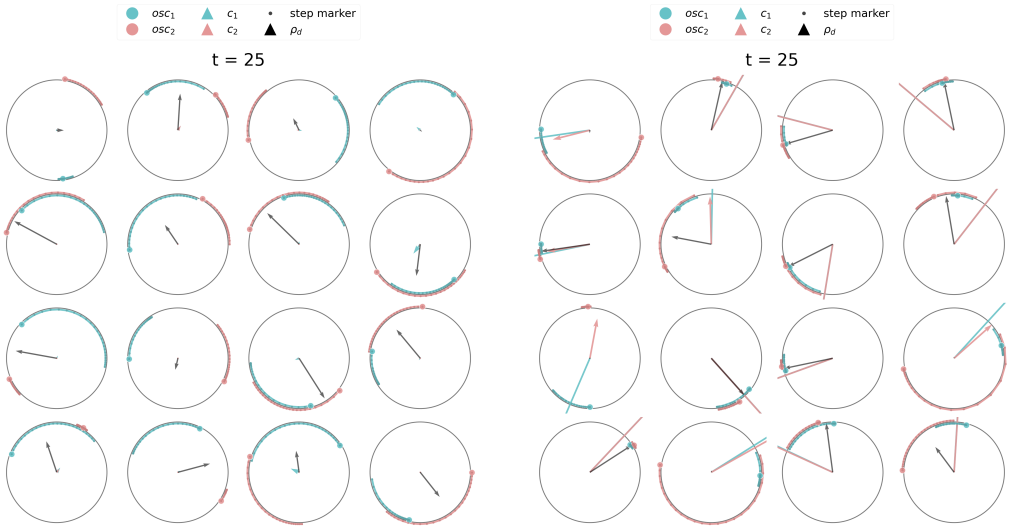
6.4 Phase–Space Visualization

The energy analysis in the previous subsection establishes that, with symmetric couplings and $\mathbf{\Omega} = 0$, the network converges to low–energy attractors that separate *same* from

different inputs. A complementary question is *how* that separation is manifested in the phase space of the oscillators themselves. Accordingly, we plot each corresponding pair of feature oscillators on a unit circle and trace their trajectories over the 25 discrete Kuramoto steps executed by the network. Phase portraits for one representative *different* pair and one representative *same* pair are shown in Figures 8 and 9 respectively.



(a) Example *different* pair



(b) Oscillator states at initialization

(c) Oscillator states after training (epoch 50)

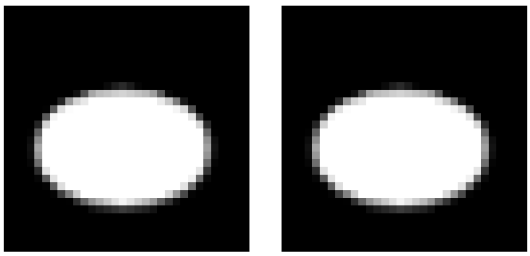
Figure 8: Oscillator dynamics for a *different* pair at timestep $T = 25$.

At random initialization the two objects in both settings are encoded by unstructured phase configurations: "same" and "different" oscillators are scattered around the circle. The conditional-stimulus vectors $\mathbf{c}_1$ and $\mathbf{c}_2$, plotted as small triangles, are near zero, confirming that the feature extractor has not yet learned a meaningful alignment between pixel intensities and oscillator phases. Consequently the coherence values ρ_d, depicted by the black arrows measuring the synchronization of each corresponding pair of oscillators, vary greatly in length and exhibit no systematic relation to the ground-truth labels.

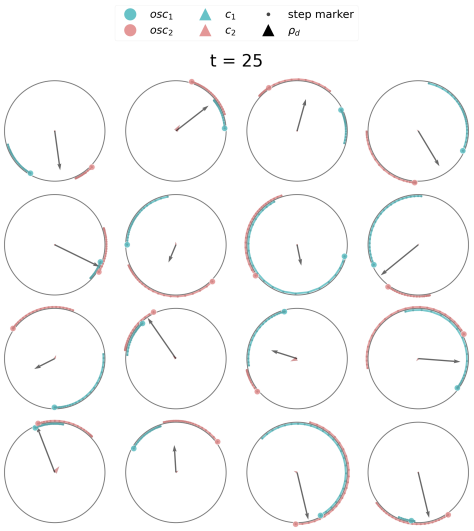
After training, the oscillator states change qualitatively. In the *same* pair the two phase clouds nearly collapse onto one another: for the majority of dimensions the cyan and red trajectories spiral into a shared fixed point and the associated coherence arrows lengthen toward the theoretical maximum. This global phase locking is exactly the signature predicted by the binding-by-synchrony hypothesis and explains why the coherence bottleneck supplies the downstream classifier with an easily separable representation of sameness.

In the *different* trial, by contrast, the system adopts a "hybrid" regime. A subset of dimensions—likely those corresponding to low-level visual regularities—still converge to a common phase, but some others settle into antiphase or remain desynchronized (with

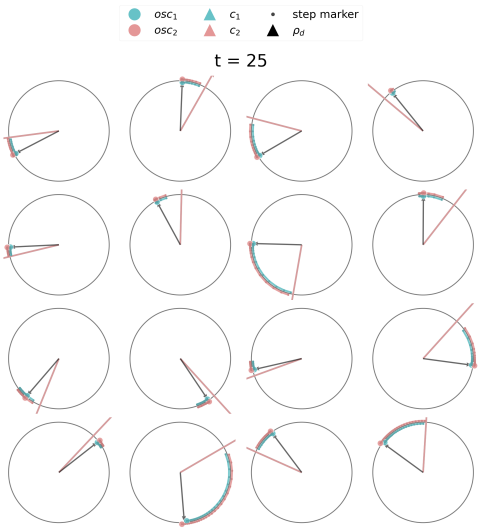
corresponding coherence measurements closer to zero). The resulting mixture yields a lower aggregate coherence, consistent with the higher final energy documented earlier and with the classifier's assignment of the "different'' label.



(a) Example *same* pair



(b) Oscillator states at initialization



(c) Oscillator states after training (epoch 50)

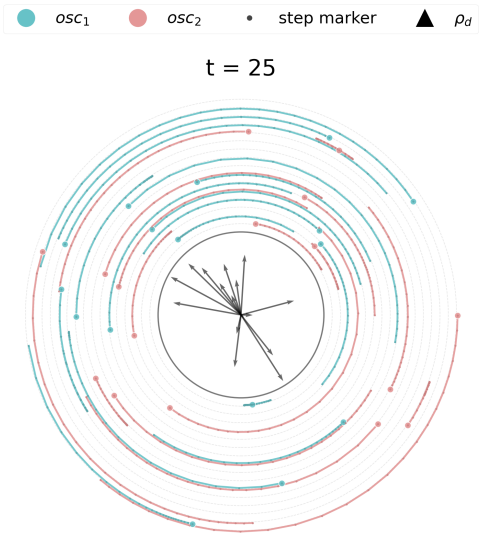
Figure 9: Oscillator dynamics for a *same* pair at timestep $T = 25$.

Crucially, these phase portraits demonstrate that the relational information is *not* stored in any single oscillator. Instead, it emerges from the collective pattern of which dimensions lock and which remain free, a pattern created by the learned coupling matrices. The model therefore realizes the relational bottleneck proposed in Section 2.2: synchrony acts as a data-dependent filter that amplifies structure-relevant phase relationships while suppressing object-specific noise. The coherence vector captures that filtered signal in compressed form, enabling the rapid transfer from Same/Different to RMTS observed in Section 6.1. Phase-space inspection thus provides a mechanistic bridge between the low-energy attractors identified in the previous subsection and the behavioral gains reported throughout Section 6.

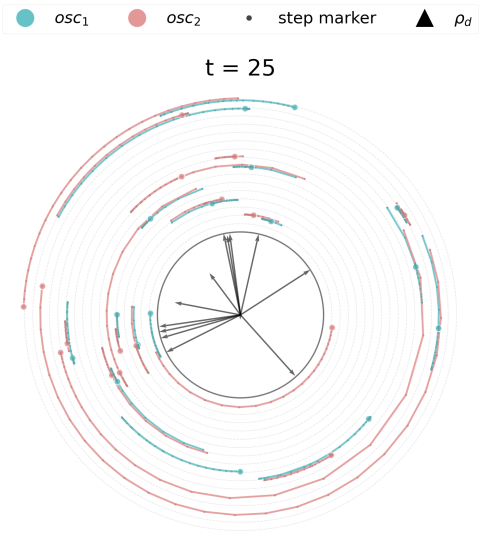
6.5 Unified Phase Portrait

The dimension–wise phase portraits of the preceding subsection revealed that individual feature oscillators either converge or repel according to the relational label. Here we ask whether these local interactions self-organize into a coherent population-level pattern. To that end we place every pair on a single, *shared* unit circle and apply a small artificial radial offset to each dimension so that individual trajectories remain visible. The resulting plots (Figure 10) provide a global view of the network's internal state before and after learning.

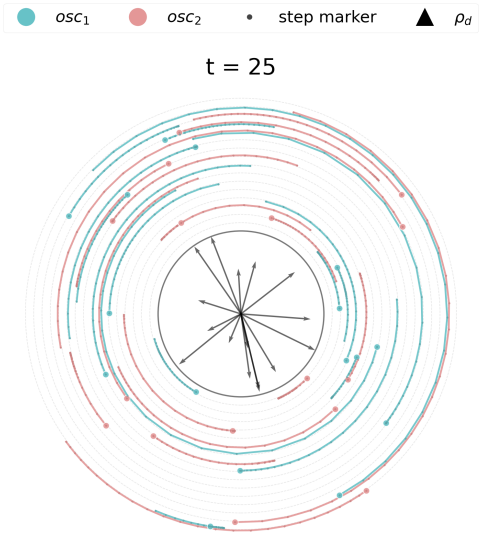
Again, at random initialization the *same* and *different* pairs are virtually indistinguishable. Cyan and red points lie at all angles; their trajectories are produced by the



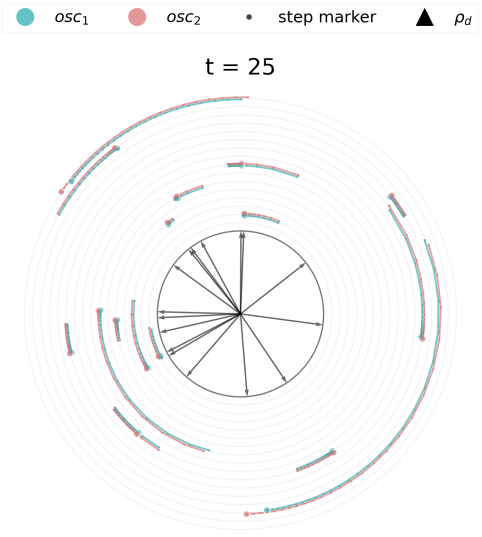
(a) *Different* pair at initialization



(b) *Different* pair after training (epoch 50)



(c) *Same* pair at initialization



(d) *Same* pair after training (epoch 50)

Figure 10: Unified portrait of oscillator phases.

randomly sampled couplings; and their phase interplay forms no discernible structure. The system therefore begins in a high-entropy regime in which each feature dimension evolves almost independently of the others.

After learning, the patterns noticeably diverge. In the *same* condition the endpoint of almost every cyan–red pair lies at the *same* angle, even though those angles span the full circle. In other words, the absolute phase of each dimension remains free, but the *relative* phase between the two objects collapses to (approximately) zero across the entire feature set. This uniform cross-object alignment constitutes global phase locking and drives the coherence vector toward its maximum, which explains the energy curve disparities and the classifier performance described earlier.

The *different* condition displays a mixed regime. A subset of dimensions—again, likely those dominated by low-level regularities that both characters share—still exhibits near-zero phase difference, but some others remain de-synchronized. The inner arrows for those dimensions are correspondingly short or opposed. Accordingly, the phase portrait consists of wedges of high synchrony interleaved with some wedges of pronounced de-synchrony. With respect to energy, this configuration reflects the higher energy state predicted by the class-conditioned Lyapunov curves (as any de-synchronized oscillator pairs increase the energy's coupling term). Functionally, the configuration realizes the relational bottleneck by decorrelating object-specific features while synchronizing abstractly relevant ones.

These unified portraits therefore provide further evidence for binding-by-synchrony. Training molds the coupling matrices so that "same" inputs drive *all* corresponding oscillators into a common phase, while "different" pairs leave the population split into several loosely coupled phase clusters. In effect, the initial input pixels are re-encoded as a geometric pattern in phase space whose overall shape distinguishes "same'' from "different''.

7 Conclusion

In this work, we introduced a neural architecture that leverages oscillatory dynamics for relational reasoning, and we demonstrated that *binding-by-synchrony* is an effective computational mechanism for structured cognition. Our model's empirical findings show that dynamic phase synchronization can serve as a robust encoding of abstract relations: for example, on the same/different task, the network reliably synchronizes the oscillatory units for "same" pairs and de-synchronizes them for "different" pairs, mirroring the neuroscience hypothesis that neurons representing a coherent percept fire in unison. This outcome unifies our results with theoretical and neuroscientific foundations: the model operationalizes the long-standing idea from neuroscience that synchronous oscillations can flexibly bind features or entities into relational structures. By integrating this principle into a learnable neural network, we bridge symbolic relational reasoning and biologically-inspired dynamics.

The contribution of our model is thus twofold. First, it provides a proof-of-concept that oscillatory-phase coding can perform complex relational computations within a neural architecture, moving beyond static feature binding to relational abstraction. The network learns an abstract notion of relations that generalizes across different feature values, indicating that it captures the essence of the relation rather than memorizing superficial patterns. Second, our approach offers notable advantages in performance and interpretability. Empirically, the oscillatory model achieves accuracy on par with or superior to our baselines on the evaluated tasks, suggesting that the inclusion of dynamic binding mechanisms does not hinder learning and can even improve it. Simultaneously, the internal oscillation patterns give rise to interpretable representations of the network's reasoning process. For instance, one can literally observe the network's decision by inspecting phase coherence between units—a property that most conventional neural networks do not readily afford. This interpretability, grounded in the model's design, provides insight into how the solution is reached (e.g., synchronized phases indicating a perceived match), thereby opening the black box of relational reasoning to some degree.

In summary, our findings validate the effectiveness of oscillatory dynamics as a medium for relational reasoning in neural systems. The model marries neuroscientific inspiration (dynamic binding by synchrony) with modern neural computation, yielding a system that not only performs well but also aligns with how brains might solve the binding problem. We believe this work lays a foundation for dynamical neural networks that inherently represent relations, contributing a new perspective to the literature on relational reasoning. By demonstrating that oscillation-based binding can achieve relational abstraction, we provide a novel avenue for building more interpretable and theoretically grounded artificial intelligence systems capable of structured cognition.

8 Limitations and Future Directions

While our results are promising, several limitations of the present work suggest clear paths for future investigation.

First, we rely on phase synchronization as the sole measure of binding; exploring alternative coherence measures (for example, frequency-based metrics or cross-frequency coupling) is an important next step. Different aspects of oscillatory activity (such as oscillation frequency or amplitude modulation) could carry additional information, and examining these might reveal whether relations can be encoded in other dynamic signatures beyond phase alignment. A frequency-based binding scheme, for instance, could allow multiple distinct relations to be represented simultaneously by assigning different frequency bands to different bindings.

Second, our current coupling mechanism uses a general inter-feature coupling matrix $J_{d,d'}$ that links *any* feature dimensions of the objects. An open question is whether focusing on intra-dimensional coupling ($J_{d,d}$ only) might suffice or even perform better. In other words, if the model only synchronizes corresponding features (e.g., color with color, shape with shape) between objects, would it still learn the relation as effectively? Investigating this could clarify the role of cross-dimensional interactions in the binding process. It may turn out that restricting coupling to within the same feature type simplifies learning and improves interpretability (by explicitly binding like features), or conversely, that the flexibility of general $J_{d,d'}$ is crucial for capturing more complex relations.

Third, we have so far evaluated the model on relatively simple relational tasks (primarily the same/different discrimination). It remains to be seen how the approach scales to more complex relational structures such as the distribution-of-three task or identity-rule tasks described in Webb et al. (2021). These problems involve multiple entities and more intricate rules (e.g., ensuring a set of three objects each has a unique value on a certain attribute, or detecting an identity match within a sequence), which pose a greater challenge for binding mechanisms. Testing our oscillatory architecture on such tasks will further assess its generality and robustness. Success on distribution-of-three or identity-rule problems would demonstrate that the model can represent and reason about higher-order relations and not just pairwise similarities, whereas any failure or difficulty would highlight the current limits of the dynamic binding approach and guide necessary refinements (such as adding more oscillatory units or hierarchical binding strategies).

Fourth, an exciting extension would be to adapt the model for sequence-to-sequence relational comparisons, for example evaluating analogical reasoning problems where two sequences of objects must be compared (e.g., 'A is to B as C is to D'). Handling two paired relations in parallel would require the network to form simultaneous bindings (one for the relation in the first pair, another for the second pair) and then to compare those bindings. Our present architecture, which binds a single pair of objects at a time, might be expanded (perhaps by an additional layer of oscillators) to juggle multiple relational bindings and determine if they are equivalent. Implementing and testing such a sequence-to-sequence comparison would push the model towards more realistic reasoning tasks and help identify if any modifications (like distinct frequency channels for different pairs, or a sequential binding process) are needed to maintain performance in analogical reasoning scenarios.

Finally, we see opportunities to integrate the oscillatory representation scheme with more powerful neural reasoning models, such as the Abstractor transformer (Altabaa et al. (2024)) or other relational architectures. In future work, one could embed our oscillatory binding module into a transformer network that has a dedicated relational reasoning inductive bias (for instance, the Abstractor is designed to explicitly handle role-binding and symbolic structure). By doing so, we could combine the strengths of both approaches: the transformer's capacity to handle complex, multi-step reasoning and generalization, with the interpretable dynamic binding that our oscillatory units provide. Such a hybrid model might, for example, use phase-synchronized oscillator groups to represent entity bindings at the input or intermediate stages, and then let the transformer layers operate over these structured representations. We anticipate that this synergy could improve performance on complex reasoning benchmarks and simultaneously yield networks whose internal logic is more transparent, thanks to the oscillatory signals.

In summary, addressing these limitations — exploring new coherence measures, refining the coupling schema, tackling more complex relational tasks, extending to sequential analogies, and merging with advanced architectures — will not only test the limits of our current model but also guide the development of next-generation relational reasoning systems. Each direction offers a chance to deepen our understanding of how oscillatory dynamics can support abstract reasoning, and together they will help assess the full potential of binding-by-synchrony principles in modern artificial intelligence models.

A Code Availability

The source code used to implement the oscillatory binding architecture, reproduce all experiments, and generate the figures reported in this paper is openly available at:

`https://github.com/Elder453/osc-network`.

References

Altabaa, Awni, Taylor Webb, Jonathan Cohen and John Lafferty. 2024. Abstractors and relational cross-attention: An inductive bias for explicit relational reasoning in transformers. URL https://arxiv.org/abs/2304.00195.

Aoyagi, Toshio. 1995. Network of neural oscillators for retrieving phase information. *Physical review letters* 74. 4075–4078.

Barrett, David G. T., Felix Hill, Adam Santoro, Ari S. Morcos and Timothy Lillicrap. 2018. Measuring abstract reasoning in neural networks. URL https://arxiv.org/abs/1807.04225.

Gentner, Dedre. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science* 7(2). 155–170.

Heusser, Andrew C, David Poeppel, Youssef Ezzyat and Lila Davachi. 2016. Episodic sequence memory is supported by a theta-gamma phase code. *Nature Neuroscience* 19(10). 1374–1380.

Hummel, John E and Keith J Holyoak. 2003. A symbolic-connectionist theory of relational inference and generalization. *Psychological Review* 110(2). 220–264.

Kim, Junkyung, Matthew Ricci and Thomas Serre. 2018. Not-so-clevr: learning same–different relations strains feedforward neural networks. *Interface Focus* 8(4). 20180011. doi:10.1098/rsfs.2018.0011. URL http://dx.doi.org/10.1098/rsfs.2018.0011.

Kuramoto, Yoshiki. 1975. Self-entrainment of a population of coupled non-linear oscillators. In Huzihiro Araki (ed.), *International symposium on mathematical problems in theoretical physics*, 420–422. Berlin, Heidelberg: Springer Berlin Heidelberg.

Lake, Brenden M., Tomer D. Ullman, Joshua B. Tenenbaum and Samuel J. Gershman. 2017. Building machines that learn and think like people. *Behavioral and Brain Sciences* 40. e253. doi:10.1017/S0140525X16001837.

Lisman, John E. and Ole Jensen. 2013. The theta-gamma neural code. *Neuron* 77(6). 1002–1016.

von der Malsburg, Ch. and W. Schneider. 1986. A neural cocktail-party processor. *Biological Cybernetics* 54(1). 29–40.

Miyato, Takeru, Sindy Löwe, Andreas Geiger and Max Welling. 2025. Artificial kuramoto oscillatory neurons. URL https://arxiv.org/abs/2410.13821.

Premack, David. 1983. The codes of man and beasts. *Behavioral and Brain Sciences* 6(1). 125–136. doi:10.1017/S0140525X00015077.

Singer, W. 2007. Binding by synchrony. *Scholarpedia* 2(12). 1657. doi:10.4249/scholarpedia.1657. Revision #124403.

Singer, Wolf. 1999. Neuronal synchrony: A versatile code for the definition of relations? *Neuron* 24(1). 49–65.

Wang, Tianshi and Jaijeet Roychowdhury. 2017. Oscillator-based ising machine. URL https://arxiv.org/abs/1709.08102.

Webb, Taylor W., Steven M. Frankland, Awni Altabaa, Simon Segert, Kamesh Krishnamurthy, Declan Campbell, Jacob Russin, Tyler Giallanza, Zack Dulberg, Randall O'Reilly, John Lafferty and Jonathan D. Cohen. 2024. The relational bottleneck as an inductive bias for efficient abstraction. URL https://arxiv.org/abs/2309.06629.

Webb, Taylor W., Ishan Sinha and Jonathan D. Cohen. 2021. Emergent symbols through binding in external memory. URL https://arxiv.org/abs/2012.14601.