Learning Interpretable Dynamics of Temporal Networks via Neural ODEs and Symbolic Regression

Connor Smith[1] and Giulio V. Dalla Riva[1]

[1]School of Mathematics and Statistics, University of Canterbury, New Zealand

Abstract

Temporal networks—networks whose structure changes over time—appear across domains from neuroscience to ecology to social systems. While most approaches focus on predicting future network states, they rarely provide interpretable models of the underlying dynamics. We present a framework for learning continuous, interpretable differential equations governing the evolution of temporal network structure. Our approach embeds networks into a low-dimensional latent space via Random Dot Product Graphs (RDPG) and aims to learn the dynamics of this embedding using Neural Ordinary Differential Equations (Neural ODEs). We develop a gauge-theoretic analysis showing that RDPG embeddings have rotational ambiguity ($O(d)$ gauge freedom) and characterize which dynamics are observable versus invisible under this symmetry. A critical challenge is that adjacency spectral embedding (ASE) introduces arbitrary gauge transformations at each time step—random rotations from SVD sign ambiguity—that are unrelated to the dynamics and make trajectory learning ill-posed. Existing joint embedding methods (e.g., UASE) assume generative models incompatible with ODE dynamics on latent positions. We propose **structure-constrained alignment**: joint optimization over gauges and dynamics where symmetry constraints on the dynamics family identify and remove the random ASE gauge artifacts. We prove that symmetric dynamics cannot fit the skew-symmetric contamination from wrong gauges, enabling recovery of the true trajectory. This yields gauge-consistent architectures $\dot{X} = N(P)X$ with symmetric N, achieving dramatic parameter reduction. We present mathematical foundations, algorithms with identifiability guarantees, and honest assessment of remaining challenges.

1 Introduction

Temporal networks—networks whose edges and nodes change over time—are ubiquitous in complex systems [1]. Examples include protein interaction networks that rewire during cellular processes [2], social networks where relationships form and dissolve [3], and ecological networks whose structure responds to environmental change [4]. Understanding how and why network structure changes is central to predicting system behavior.

Most temporal network modeling falls into two categories. The first models *dynamics on networks*: how node states evolve given a fixed or slowly-changing network topology (e.g., epidemic spreading, opinion dynamics) [5]. The second models *dynamics of networks*: how the network structure itself evolves [6]. This paper addresses the latter, which remains less developed despite its importance.

Existing approaches to modeling network dynamics face a fundamental tension. Statistical models like temporal exponential random graphs [3] are interpretable but often lack predictive power. Machine learning approaches [7] achieve better predictions but function as black boxes, offering little insight into the mechanisms driving structural change. We propose a framework that achieves both: predictive models that can be distilled into interpretable differential equations.

Our key insight is that the discreteness of network events (edges appearing or disappearing) can be overcome by working in a continuous embedding space. We use Random Dot Product Graphs (RDPG) [8] to embed each network snapshot into a low-dimensional latent space where similar nodes cluster together and connection probabilities arise naturally from inner products. The temporal evolution of these embeddings is then smooth and amenable to differential equation modeling.

We train Neural Ordinary Differential Equations (Neural ODEs) [9] to learn the dynamics in embedding space. While Neural ODEs provide excellent fits, they remain opaque. We therefore apply symbolic regression to discover closed-form differential equations that approximate the learned neural dynamics. These equations are interpretable—they can be analyzed mathematically, checked for conservation laws, and compared across systems.

Contributions. We make contributions in three areas:

Theoretical foundations:

1. A gauge-theoretic analysis of RDPG dynamics, characterizing observable vs. invisible dynamics and deriving gauge-invariant parameterizations ($\dot{X} = N(P)X$ with N symmetric)
2. The horizontal lift framework: fiber bundle geometry for understanding what "gauge-free trajectory" means, connecting to Procrustes alignment
3. Identifiability results: conditions under which dynamics parameters can be recovered from observed probability matrices

Critical analysis:

1. Demonstration that existing joint embedding methods (UASE, Omnibus) assume generative models incompatible with ODE dynamics on latent positions
2. Honest assessment of gauge alignment as a hard open problem for general dynamics
3. Clarification of what spectral methods can and cannot provide

Proposed approach:

1. **Structure-constrained alignment**: joint optimization over gauges and dynamics, using structural assumptions (symmetry, sparsity) as regularization
2. Alternating algorithm for linear horizontal dynamics, with identifiability guarantees under generic conditions
3. Discussion of extensions to polynomial and message-passing dynamics

2 Methods

Our framework consists of three stages: (1) embedding temporal networks via RDPG, (2) learning dynamics with Neural ODEs, and (3) extracting interpretable equations through symbolic regression.

2.1 Random Dot Product Graphs and Latent Position Space

We begin by establishing the latent space framework for Random Dot Product Graphs (RDPG) [8], [10].

2.1.1 Latent positions and connection probabilities

In an RDPG, each node i is associated with a latent position that determines its propensity to form connections. For directed graphs, this position has two components:

- A **source position** $\vec{g}_i \in B^d_+$: the node's propensity to initiate connections
- A **target position** $\vec{r}_i \in B^d_+$: the node's propensity to receive connections

Here B^d_+ denotes the positive orthant of the unit ball:

$$B^d_+ = \{x \in \mathbb{R}^d : x_k \geq 0 \text{ for all } k, \quad \|x\| \leq 1\} \tag{1}$$

The probability of a directed edge from node i to node j is given by the dot product:

$$P_{ij} = \vec{g}_i \cdot \vec{r}_j \tag{2}$$

The constraint $\vec{g}_i, \vec{r}_j \in B^d_+$ ensures $P_{ij} \in [0, 1]$: non-negativity of coordinates gives $P_{ij} \geq 0$, while Cauchy-Schwarz gives $P_{ij} \leq \|\vec{g}_i\| \cdot \|\vec{r}_j\| \leq 1$.

More generally, the RDPG literature defines an **inner product distribution** F as any distribution on $\mathbb{R}^d$ such that $\vec{x}_i^\top \vec{x}_j \in [0, 1]$ for all pairs drawn from F [8]. The positive orthant constraint B^d_+ is a convenient sufficient condition, but other configurations (e.g., positions on the Hardy-Weinberg curve in population genetics) also yield valid probability matrices.

Collecting positions into matrices $G, R \in \mathbb{R}^{n \times d}$ with rows $G_{i\cdot} = \vec{g}_i^\top$ and $R_{i\cdot} = \vec{r}_i^\top$, the probability matrix is:

$$P = GR^\top \tag{3}$$

For **undirected** graphs, we impose $\vec{g}_i = \vec{r}_i =: \vec{x}_i$, so that $P_{ij} = \vec{x}_i \cdot \vec{x}_j$ is symmetric. Writing $X \in \mathbb{R}^{n \times d}$ for the matrix with rows $\vec{x}_i^\top$:

$$P = XX^\top \tag{4}$$

We develop the theory primarily for the undirected case, noting that the directed generalization $(G, R) \mapsto GR^\top$ follows similar principles (see Section 10).

2.1.2 Adjacency spectral embedding

Given an observed adjacency matrix $A \in \{0, 1\}^{n \times n}$, we estimate latent positions via the **adjacency spectral embedding** (ASE): a truncated singular value decomposition of A. For $A = U\Sigma V^\top$, the rank-d approximation gives:

$$\hat{G} = U_d \Sigma_d^{\frac{1}{2}}, \quad \hat{R} = V_d \Sigma_d^{\frac{1}{2}} \tag{5}$$

For undirected graphs with symmetric A, we have $U = V$ and set $\widehat{X} = U_d \Sigma_d^{\frac{1}{2}}$.

The estimated positions $\widehat{X}$ may not lie exactly in $\left(B_+^d\right)^n$, but for well-specified models they concentrate near the true positions [8]. This spectral approach is consistent: as $n \to \infty$, $\widehat{X}$ converges to the true positions up to orthogonal transformation.

2.1.3 Temporal alignment via Procrustes

For a temporal network $\{A_t\}_{t=1}^T$, we embed each snapshot independently, then align across time. SVD determines positions only up to orthogonal transformation: if $\widehat{X}$ is a valid embedding, so is $\widehat{X}Q$ for any $Q \in O(d)$. This nonidentifiability is fundamental to the RDPG model [8].

We align consecutive embeddings using the **orthogonal Procrustes** solution:

$$Q_t = \arg \min_{Q^\top Q = I} \|\widehat{X}_t Q - \widehat{X}_{t-1}\|_F \tag{6}$$

which has closed form $Q_t = V U^\top$ where $\widehat{X}_t^\top \widehat{X}_{t-1} = U \Sigma V^\top$.

2.1.4 The alignment problem for learning dynamics

While pairwise Procrustes alignment produces reasonable discrete approximations, it is fundamentally inadequate for learning continuous dynamics. The problem is more severe than simple "drift"—the gauge can jump discontinuously even for arbitrarily small time steps.

The discrete gauge ambiguity. At each time t, ASE computes the eigendecomposition $P(t) \approx U(t)\Lambda(t)U(t)^\top$ and returns $\widehat{X}(t) = U(t)\Lambda(t)^{1/2}$. But eigenvectors are determined only up to sign: if u is an eigenvector, so is $-u$. For embedding dimension d, this gives 2^d possible gauge choices at each time, and the SVD algorithm picks arbitrarily based on numerical details.

Consequence: gauge jumps are $O(1)$ even for infinitesimal δt. Suppose $P(t)$ and $P(t + \delta t)$ differ by $O(\delta t)$. The eigenvalues vary continuously, but the eigenvector signs can flip between consecutive times. When this happens, $\widehat{X}(t + \delta t)$ and $\widehat{X}(t)$ differ by a reflection—an $O(1)$ change—even though the true dynamics moved by $O(\delta t)$.

This means we cannot estimate velocities by finite differences:

$$\frac{\widehat{X}(t + \delta t) - \widehat{X}(t)}{\delta t} \tag{7}$$

is not approximating any smooth velocity. It's dominated by the gauge jump, which contributes $O(1/\delta t) \to \infty$ as $\delta t \to 0$.

Pairwise Procrustes doesn't solve this. Aligning $\widehat{X}(t + 1)$ to $\widehat{X}(t)$ via Procrustes finds the best orthogonal transformation *between those two frames*. But:

- Each pairwise alignment $Q_{t,t+1}$ can be a large rotation (reflecting sign flips)
- The composition $Q_{0,1} Q_{1,2} \cdots Q_{T-1,T}$ accumulates these rotations
- The resulting trajectory, while locally aligned, can wander far from horizontal

In short: **we don't have a scrambled trajectory—we don't have a trajectory at all**. We have a sequence of snapshots, each in an arbitrary gauge unrelated to its neighbors.

2.1.5 Why existing joint embedding methods don't solve this

A natural question is whether existing multi-graph embedding methods resolve the gauge alignment problem. The most prominent is **Unfolded Adjacency Spectral Embedding** (UASE) [11], which embeds all time points jointly via a single SVD.

> **Definition 1** (UASE). Given adjacency matrices $A^{(1)}, ..., A^{(T)} \in \{0,1\}^{n \times n}$:
> 1. Form the unfolded matrix $\boldsymbol{A} = \left(A^{(1)} \mid \cdots \mid A^{(T)}\right) \in \mathbb{R}^{n \times nT}$
> 2. Compute the rank-d SVD: $\boldsymbol{A} \approx \widehat{X}\hat{Y}^\top$ where $\widehat{X} \in \mathbb{R}^{n \times d}$, $\hat{Y} \in \mathbb{R}^{nT \times d}$
> 3. Partition $\hat{Y} = \left(\hat{Y}^{(1)}; \cdots; \hat{Y}^{(T)}\right)$ into T blocks

UASE produces gauge-consistent embeddings—all $\hat{Y}^{(t)}$ share a common basis—and satisfies important stability properties [11]. However, **UASE assumes a different generative model than ours**.

The model mismatch. UASE is designed for the *Multilayer RDPG* where:

$$P_{ij}^{(t)} = X_i \Lambda^{(t)} Y_j^{(t)\top} \tag{8}$$

with X **constant** across time and only $Y^{(t)}$ varying. This factored structure is appropriate when nodes have a stable "identity" component and time-varying "activity" patterns.

Our dynamics model is different:

$$P_{ij}^{(t)} = x_i(t)^\top x_j(t) \tag{9}$$

where the **same** evolving position $x_i(t)$ determines both sending and receiving, and the positions themselves follow ODE dynamics. There is no factorization through a shared constant component.

Consequence: UASE distorts the dynamics. When applied to data from $P^{(t)} = X(t)X(t)^\top$ with genuinely evolving $X(t)$, UASE finds a compromise:

- $\widehat{X}$ captures some "average" structure across time
- $\hat{Y}^{(t)}$ absorbs temporal variation, but in a way that doesn't correspond to true position evolution

In practice, UASE can produce trajectories that fail to capture the actual dynamics. The joint SVD enforces a low-rank factored structure that the true model doesn't satisfy.

> *Remark.* Similar limitations apply to Omnibus embedding [12], which builds a block matrix of averaged adjacencies, and to COSIE [13]. All assume some shared structure across time that our ODE-driven dynamics model lacks.

What we need: A global gauge-fixing procedure that:

1. Works with independent ASE at each time (respecting our generative model)
2. Finds gauges $Q_t \in O(d)$ for all times simultaneously
3. Produces a genuinely smooth trajectory $\widetilde{X}(t) = \widehat{X}(t)Q_t$
4. Approximates the horizontal lift (minimal gauge motion)

2.1.6 Fiber bundle geometry and horizontal lifts

To understand what "smooth" means for gauge-ambiguous data, we formalize the geometry. The embedding space forms a **principal fiber bundle** with structure group $O(d)$.

> **Definition 2** (Latent Position Bundle). Let $\mathcal{M} = \{X \in \mathbb{R}^{n \times d} : \operatorname{rank}(X) = d\}$ be the space of full-rank latent position matrices. The **projection** $\pi : \mathcal{M} \to \mathcal{P}$ maps $X \mapsto P = XX^\top$ to the space of rank-d positive semidefinite matrices. The **fiber** over P is $\pi^{-1}(P) = \{XQ : Q \in O(d)\}$ —the gauge orbit.

A path $P(t)$ in the base space (observable dynamics) can be lifted to many paths $X(t)$ in the total space. These lifts differ by time-dependent gauge transformations $Q(t) \in O(d)$. The key question: among all lifts, which is the "natural" one?

> **Definition 3** (Vertical and Horizontal Subspaces). At each $X \in \mathcal{M}$, the tangent space decomposes as $T_X\mathcal{M} = \mathcal{V}_X \oplus \mathcal{H}_X$ where:
>
> - **Vertical subspace** (gauge directions): $\mathcal{V}_X = \{XA : A \in \mathfrak{so}(d)\}$
> - **Horizontal subspace** (observable directions): $\mathcal{H}_X = \mathcal{V}_X^\perp$
>
> A tangent vector $\dot{X}$ is **horizontal** if $\dot{X} \in \mathcal{H}_X$, i.e., it has no gauge component.

Proposition 1 (Horizontal Characterization). A velocity $\dot{X}$ is horizontal if and only if $X^\top \dot{X}$ is symmetric. Equivalently, $\dot{X}^\top X = X^\top \dot{X}$.

Proof. Decompose $\dot{X} = XA + W$ where $A = \left(X^\top X\right)^{-1} X^\top \dot{X}$ and $W \perp \mathsf{col}(X)$. The vertical component is XA_{skew} where $A_{\text{skew}} = \frac{A - A^\top}{2}$. Thus $\dot{X} \in \mathcal{H}_X$ iff $A_{\text{skew}} = 0$ iff $A = A^\top$ iff $X^\top \dot{X}$ is symmetric.□

> **Definition 4** (Horizontal Lift). Given a path $P(t)$ in the base space and initial condition $X(0)$ with $X(0)X(0)^\top = P(0)$, the **horizontal lift** is the unique path $X(t)$ satisfying:
>
> 1. $X(t)X(t)^\top = P(t)$ for all t (projects to P)
> 2. $\dot{X}(t) \in \mathcal{H}_{X(t)}$ for all t (velocity is horizontal)

The horizontal lift is the "smoothest" lift in a precise sense: it has no superfluous gauge motion. This is exactly what we want for learning dynamics.

Theorem 1 (Horizontal Lift Existence and Uniqueness). Let $P(t)$ be a smooth path of rank-d positive semidefinite matrices. For any initial X_0 with $X_0 X_0^\top = P(0)$, there exists a unique horizontal lift $X(t)$ with $X(0) = X_0$.

Proof. Differentiating $P = XX^\top$ gives $\dot{P} = \dot{X}X^\top + X\dot{X}^\top$. For $\dot{X}$ horizontal, we require $X^\top \dot{X} = S$ symmetric. Substituting $\dot{X} = X\left(X^\top X\right)^{-1} S + W$ with $X^\top W = 0$:

$$\dot{P} = X\left(X^\top X\right)^{-1} S X^\top + X S \left(X^\top X\right)^{-1} X^\top + W X^\top + X W^\top \tag{10}$$

Given $\dot{P}$ and requiring $WX^\top + XW^\top$ to match the residual, this is a system of linear equations for (S, W). For full-rank X, standard ODE existence/uniqueness theory applies. □

2.1.7 Global gauge synchronization

The horizontal lift theorem tells us what we want; now we need a practical method to approximate it from discrete, gauge-scrambled embeddings.

The synchronization problem. Given embeddings $\left\{\widehat{X}_0, \widehat{X}_1, ..., \widehat{X}_T\right\}$ with arbitrary gauges, find gauge corrections $\{Q_0, Q_1, ..., Q_T\} \subset O(d)$ such that the corrected trajectory $\widetilde{X}_t = \widehat{X}_t Q_t$ is as smooth as possible.

This is an instance of **synchronization over groups**, a well-studied problem in computer vision and signal processing.

Definition 5 (Gauge Synchronization Objective). The **smoothness energy** of a gauge assignment $\{Q_t\}$ is:

$$\mathcal{E}(\{Q_t\}) = \sum_{t=0}^{T-1} \|\widehat{X}_{t+1} Q_{t+1} - \widehat{X}_t Q_t\|_F^2 \tag{11}$$

The **synchronized gauges** minimize $\mathcal{E}$ over $(O(d))^{T+1}$.

Minimizing $\mathcal{E}$ directly is nonconvex, but good algorithms exist. A key observation simplifies the problem: only relative gauges $R_t = Q_t^\top Q_{t+1}$ matter for smoothness.

Proposition 2 (Relative Gauge Formulation). Define $R_t = Q_t^\top Q_{t+1}$ (relative gauge from t to $t+1$). Then:

$$\mathcal{E} = \sum_{t=0}^{T-1} \|\widehat{X}_{t+1} R_t^\top \cdots R_0^\top Q_0 - \widehat{X}_t R_{t-1}^\top \cdots R_0^\top Q_0\|_F^2 \tag{12}$$

The optimal Q_0 is arbitrary (global gauge freedom); only the R_t affect smoothness.

Connection to Procrustes. The pairwise Procrustes solution $R_t^{\text{pair}} = \arg\min_{R \in O(d)} \|\widehat{X}_{t+1} R - \widehat{X}_t\|_F$ provides an initial estimate. But these local solutions may be inconsistent: $R_0^{\text{pair}} R_1^{\text{pair}} \cdots R_{T-1}^{\text{pair}}$ may not equal I even if the path is closed.

Spectral relaxation. A practical algorithm relaxes $O(d)$ to the Stiefel manifold and solves via eigendecomposition. Define the **connection Laplacian**:

$$L_{t,t'} = \begin{cases} I & \text{if } t = t' \\ -\widehat{X}_t^\top \widehat{X}_{t+1} \left(\widehat{X}_{t+1}^\top \widehat{X}_{t+1}\right)^{-1} & \text{if } |t - t'| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

The synchronized gauges correspond to the bottom eigenvectors of L.

Practical algorithm.

1. Compute pairwise Procrustes: $R_t^{\text{init}} = \text{Procrustes}\left(\widehat{X}_t, \widehat{X}_{t+1}\right)$
2. Chain to get initial gauges: $Q_t^{\text{init}} = R_0^{\text{init}} R_1^{\text{init}} \cdots R_{t-1}^{\text{init}}$
3. Refine via gradient descent on $\mathcal{E}$ over $(O(d))^{T+1}$, using Riemannian optimization
4. Output $\widetilde{X}_t = \widehat{X}_t Q_t$

2.1.8 The Procrustes flow (continuous limit)

The synchronization procedure above works for discrete data. In the limit of continuous observations, it converges to the **Procrustes flow**—the horizontal lift.

Theorem 2 (Procrustes Flow as Horizontal Lift). Let $\left\{\widehat{X}_t\right\}_{t\in[0,T]}$ be a continuous path of embeddings satisfying $\widehat{X}_t\widehat{X}_t^\top = P_t$ for some smooth P_t. Suppose $\widehat{X}_t = X_t R_t$ where X_t is horizontal and $R_t \in O(d)$ is the (unknown) gauge error.

Define the **gauge velocity** $\Omega_t = R_t^{-1}\dot{R}_t \in \mathfrak{so}(d)$. The Procrustes flow removes this gauge motion: if we set

$$\dot{Q} = -Q \cdot \left(\widehat{X}^\top \widehat{X}\right)^{-1} \cdot \operatorname{skew}\left(\widehat{X}^\top \dot{\widehat{X}}\right) \tag{14}$$

where $\operatorname{skew}(M) = \frac{M-M^\top}{2}$, then $\widetilde{X}_t = \widehat{X}_t Q_t$ is horizontal.

Proof. The corrected trajectory is $\widetilde{X} = \widehat{X}Q = XRQ$. Its velocity is $\dot{\widetilde{X}} = \dot{X}RQ + X\dot{R}Q + XR\dot{Q}$. For $\widetilde{X}$ to be horizontal, we need $\widetilde{X}^\top \dot{\widetilde{X}}$ symmetric.

The skew-symmetric part of $\widehat{X}^\top \dot{\widehat{X}}$ measures the gauge velocity in the raw embeddings. The flow $\dot{Q} = -Q\left(\widehat{X}^\top \widehat{X}\right)^{-1} \operatorname{skew}\left(\widehat{X}^\top \dot{\widehat{X}}\right)$ exactly cancels this, leaving only the horizontal component. □

> *Remark.* In practice, we cannot use the Procrustes flow directly because: (1) we have discrete samples, not continuous paths, and (2) the gauge jumps between samples are discontinuous, so $\dot{\widehat{X}}$ doesn't exist. The flow equation describes the **idealized** continuous-time limit; the synchronization algorithm (Section 2.1.7) provides the practical discrete-time implementation.

> *Remark.* The horizontal condition $X^\top \dot{X}$ symmetric connects directly to our invisible dynamics theorem (Theorem 4). Invisible dynamics $\dot{X} = XA$ with $A \in \mathfrak{so}(d)$ have $X^\top \dot{X} = (X^\top X)A$ skew-symmetric. The Procrustes flow/synchronization removes exactly this invisible component, leaving only observable dynamics.

> *Remark.* When the adjacency matrix A has negative eigenvalues (e.g., due to heterophilic "opposites attract" connectivity), the standard RDPG with $P = XX^\top \geq 0$ is inadequate. The **Generalized RDPG** (GRDPG) [14] handles this by using an indefinite inner product: $P_{ij} = \vec{x}_i^\top I_{p,q}\vec{x}_j$ where $I_{p,q} = \operatorname{diag}(1, ..., 1, -1, ..., -1)$. The gauge group then becomes the indefinite orthogonal group $O(p, q)$. Our gauge-theoretic analysis extends naturally to this setting, though we focus on the positive-definite case for clarity.

2.1.9 The alignment problem: an honest assessment

We must be candid: **gauge alignment from spectral embeddings alone is a hard open problem** for learning continuous dynamics. The methods described above—global synchronization, horizontal lifts, Procrustes flow—provide a theoretical framework and idealized algorithms, but face fundamental challenges in practice.

The core difficulty is not gauge drift from dynamics (which may be zero for horizontal dynamics). The problem is that **ASE introduces arbitrary gauge transformations at each time step**:

$$\widehat{X}^{(t)} = X^{(t)} R^{(t)} \tag{15}$$

where the $R^{(t)} \in O(d)$ are essentially random—determined by SVD sign conventions, not by physics. Even perfectly smooth true dynamics produce jagged, discontinuous embedding trajectories.

Specific challenges:

1. **Discontinuous gauge jumps**: The $R^{(t)}$ can change arbitrarily between consecutive time steps, making finite differences meaningless.
2. **No direct observability**: We never observe $R^{((t)\}}$ or $X^{((t)\}}$ separately—only their product $\widehat{X}^{(t)}$.
3. **Statistical noise**: ASE estimates have $O(1/\sqrt{n})$ errors that compound with alignment errors.

Existing joint embedding methods like UASE solve a different problem (the multilayer RDPG) and, as discussed in Section 2.1.5, assume a generative model incompatible with the ODE dynamics we aim to learn.

This motivates a different approach: rather than aligning first and learning dynamics second, we propose **joint optimization** where structural assumptions about dynamics help identify and remove the random ASE gauge artifacts.

2.1.10 Structure-constrained gauge alignment

The core problem is not gauge drift from dynamics—horizontal dynamics produce no such drift. The problem is that **ASE introduces arbitrary gauge transformations at each time step**, unrelated to the dynamics.

The ASE gauge artifact. At each time t, ASE computes the eigendecomposition of $A^{(t)}$ (or $\hat{P}^{(t)}$). The eigenvectors are determined only up to sign (and rotation within repeated eigenspaces). This means:

$$\widehat{X}^{(t)} = X^{(t)} R^{(t)} + E^{(t)} \tag{16}$$

where $R^{(t)} \in O(d)$ is essentially **random**—determined by numerical details of the SVD algorithm, not by the dynamics. Even if the true positions $X^{(t)}$ evolve smoothly, the estimates $\widehat{X}^{(t)}$ jump erratically due to these arbitrary gauge choices.

The alignment goal. We seek gauge corrections $Q_t \in O(d)$ such that:

$$\widetilde{X}^{(t)} := \widehat{X}^{(t)} Q_t \approx X^{(t)} Q^* \tag{17}$$

for some fixed (unknown) global gauge Q^*. That is, we want to "undo" the random ASE gauges $R^{(t)}$, recovering a trajectory that follows the true dynamics up to an overall rotation.

Why structure helps. Without additional information, the problem is underdetermined: any smooth interpolation through the $\widehat{X}^{((t)\}}$ could be "correct." But if we know (or assume) the dynamics belong to a restricted family $\mathcal{F}$, this provides a constraint. The corrected trajectory $\widetilde{X}^{(t)}$ should be **explainable** by some $f \in \mathcal{F}$. Random ASE gauges produce trajectories that require dynamics outside $\mathcal{F}$; correct gauges produce learnable trajectories.

> **Definition 6** (Joint Alignment-Learning Problem). Given ASE embeddings $\left\{\widehat{X}^{(t)}\right\}_{t=0}^{T}$ and dynamics family $\mathcal{F}$, find gauge corrections $\{Q_t \in O(d)\}$ and $f \in \mathcal{F}$ minimizing:
>
> $$\mathcal{L}(\{Q_t\}, f) = \sum_{t=0}^{T-1} \|\widehat{X}^{(t+1)} Q_{t+1} - \widehat{X}^{(t)} Q_t - \delta t \cdot f\left(\widehat{X}^{(t)} Q_t\right)\|_F^2 \tag{18}$$

The dynamics family $\mathcal{F}$ acts as regularization: it couples gauge choices across time by requiring the corrected trajectory to be **learnable** within $\mathcal{F}$.

The tradeoff:
- $\mathcal{F}$ too restrictive: true dynamics may not fit, alignment fails
- $\mathcal{F}$ too expressive (e.g., unconstrained neural network): can fit any trajectory, no constraint on gauges

We now work out the mathematics for a concrete, tractable case.

2.1.10.1 Linear horizontal dynamics

Consider the family of linear dynamics with symmetric coefficient:

$$\mathcal{F}_{\text{lin}} = \left\{\dot{X} = NX : N \in \mathbb{R}^{n \times n}, N = N^\top\right\} \tag{19}$$

The symmetry constraint $N = N^\top$ ensures dynamics are **horizontal**—purely observable, with no invisible gauge component (connecting to Theorem 4 and Proposition 1).

Discrete-time formulation. For small time step δt, the dynamics $\dot{X} = NX$ discretize to:

$$X^{(t+1)} \approx (I + \delta t \cdot N) X^{(t)} = M X^{(t)} \tag{20}$$

where $M = I + \delta t \cdot N$ inherits symmetry from N.

The observation model. True positions evolve as $X^{(t+1)} = M X^{(t)}$ for some fixed symmetric M. Note: because M is symmetric, the true dynamics are **horizontal**—there is no gauge drift from the dynamics themselves.

ASE gives us $\widehat{X}^{(t)} = X^{(t)} R^{(t)}$ where $R^{(t)} \in O(d)$ are arbitrary (from SVD sign choices). These $R^{(t)}$ are unrelated across time—they are numerical artifacts, not dynamical.

The alignment goal. Find $Q_t \approx \left(R^{(t)}\right)^{-1}$ (up to a global constant) so that:

$$\widetilde{X}^{(t)} = \widehat{X}^{(t)} Q_t \approx X^{(t)} \tag{21}$$

In other words, undo the random ASE gauges to recover the true smooth trajectory.

The optimization problem. Find gauges $\{Q_t\}$ and symmetric M minimizing:

$$\mathcal{L} = \sum_{t=0}^{T-1} \|\widehat{X}^{(t+1)} Q_{t+1} - M \widehat{X}^{(t)} Q_t\|_F^2 \tag{22}$$

Proposition 3 (Relative Gauge Reduction). Define relative gauges $R_t = Q_t^\top Q_{t+1}$. The objective Equation 22 depends on $\{Q_t\}$ only through $\{R_t\}$ and a global gauge Q_0:

$$\mathcal{L} = \sum_{t=0}^{T-1} \|\widehat{X}^{(t+1)} R_t^\top - M\widehat{X}^{(t)}\|_F^2 \tag{23}$$

The global gauge Q_0 is unidentifiable (reflecting the inherent $O(d)$ freedom); only relative gauges affect the loss.

Proof. $\widehat{X}^{(t+1)} Q_{t+1} - M\widehat{X}^{(t)} Q_t = \left(\widehat{X}^{(t+1)} R_t^\top - M\widehat{X}^{(t)}\right) Q_t$. Since Q_t is orthogonal, $\|AQ_t\|_F = \|A\|_F$. □

Alternating optimization algorithm.

The non-convex problem decomposes into tractable subproblems:

Step 1: Fix $\{R_t\}$, solve for M.

This is least squares with a symmetry constraint:

$$\min_{M=M^\top} \sum_t \|\widehat{X}^{(t+1)} R_t^\top - M\widehat{X}^{(t)}\|_F^2 \tag{24}$$

Stack the data: let $Y = \left[\widehat{X}^{(1)} R_0^\top \mid \cdots \mid \widehat{X}^{(T)} R_{T-1}^\top\right]$ and $Z = \left[\widehat{X}^{(0)} \mid \cdots \mid \widehat{X}^{(T-1)}\right]$.

The unconstrained minimizer is $M^* = YZ^\top \left(ZZ^\top\right)^{-1}$ (assuming $ZZ^\top$ is invertible).

Project onto symmetric matrices:

$$M = \frac{M^* + (M^*)^\top}{2} \tag{25}$$

Step 2: Fix M, solve for $\{R_t\}$.

For each t independently, solve the orthogonal Procrustes problem:

$$\min_{R_t \in O(d)} \|\widehat{X}^{(t+1)} R_t^\top - M\widehat{X}^{(t)}\|_F^2 \tag{26}$$

Proposition 4 (Procrustes Solution for Relative Gauges). Let $A_t = \widehat{X}^{((t))^\top} M^\top \widehat{X}^{(t+1)}$ with SVD $A_t = U_t \Sigma_t V_t^\top$. Then:

$$R_t = V_t U_t^\top \tag{27}$$

Proof. The objective $\|\widehat{X}^{(t+1)} R_t^\top - M\widehat{X}^{(t)}\|_F^2$ expands to a constant minus 2 $\operatorname{tr}(R_t A_t^\top)$. Maximizing $\operatorname{tr}(R_t A_t^\top)$ over $O(d)$ is the orthogonal Procrustes problem, solved by $R_t = V_t U_t^\top$. □

The complete algorithm:

```
**Input:** Embeddings $\left\{\widehat{X}^{(t)}\right\}_{t=0}^{T}$, tolerance $\varepsilon$
**Output:** Aligned embeddings $\left\{\widetilde{X}^{(t)}\right\}$, dynamics matrix $M$
1. **Initialize:** $R_t^{(0)} \leftarrow \text{Procrustes}\left(\widehat{X}^{(t)}, \widehat{X}^{(t+1)}\right)$ for all $t$
2. **Repeat until convergence:**
    • **M-step:** $Y \leftarrow \left[\widehat{X}^{(1)}\left(R_0^{(k)}\right)^\top \mid \cdots\right]$, $Z \leftarrow \left[\widehat{X}^{(0)} \mid \cdots\right]$
    • $M^{(k+1)} \leftarrow \frac{YZ^\top(ZZ^\top)^{-1}+\left(YZ^\top(ZZ^\top)^{-1}\right)^\top}{2}$
    • **R-step:** For each $t$: compute SVD of $\widehat{X}^{((t))^\top} M^{(k+1)} \widehat{X}^{(t+1)} = U\Sigma V^\top$
    • $R_t^{(k+1)} \leftarrow VU^\top$
3. **Recover absolute gauges:** $Q_0 \leftarrow I$, $Q_{t+1} \leftarrow Q_t R_t$ for $t = 0, ..., T-1$
4. **Return:** $\widetilde{X}^{(t)} \leftarrow \widehat{X}^{(t)} Q_t$, $M$
```

Algorithm 1: Structure-constrained gauge alignment (linear dynamics)

2.1.10.2 Why the symmetry constraint helps

Without requiring $M = M^\top$, the problem decouples: each R_t could be chosen via pairwise Procrustes independently, and we'd simply fit the best (possibly asymmetric) M afterward. This recovers the naive approach that fails—it aligns consecutive frames but doesn't use the global structure to identify the random ASE gauge artifacts.

The symmetry constraint **couples gauge choices globally** and forces recovery of the true gauges. The key insight: random ASE gauge errors $R^{(t)}$ introduce apparent "gauge velocity" that is skew-symmetric, while true horizontal dynamics are symmetric. The constraint that dynamics fit symmetric N implicitly enforces $Q_t \approx \left(R^{(t)}\right)^{-1}$. We now make this precise.

Theorem 3 (Gauge Velocity Contamination). Let $X(t)$ follow true dynamics $\dot{X} = NX$ with $N = N^\top$. Let $\widetilde{X} = XS$ for some time-varying gauge error $S(t) \in O(d)$. Then the apparent dynamics in the $\widetilde{X}$ frame are:

$$\dot{\widetilde{X}} = N\widetilde{X} + \widetilde{X}\Omega \tag{28}$$

where $\Omega = S^{-1}\dot{S} \in \mathfrak{so}(d)$ is skew-symmetric.

Moreover, this can be written as $\dot{\widetilde{X}} = \widetilde{N}\widetilde{X}$ for some symmetric $\widetilde{N}$ if and only if $\Omega = 0$.

Proof. By the product rule:

$$\dot{\widetilde{X}} = \dot{X}S + X\dot{S} = NXS + X\dot{S} \tag{29}$$

Since $S \in O(d)$, we have $SS^\top = I$, so $X = \widetilde{X}S^{-1}$. Substituting:

$$\dot{\widetilde{X}} = N\widetilde{X}S^{-1}S + \widetilde{X}S^{-1}\dot{S} = N\widetilde{X} + \widetilde{X}\Omega \tag{30}$$

where $\Omega = S^{-1}\dot{S}$. Since $SS^\top = I$, differentiating gives $\dot{S}S^\top + S\dot{S}^\top = 0$, so $\dot{S}S^\top = -S\dot{S}^\top$. Thus:

$$\Omega^\top = \left(S^{-1}\dot{S}\right)^\top = \dot{S}^\top \left(S^{-1}\right)^\top = \dot{S}^\top S = -\left(\dot{S}S^\top\right)^\top S = -S^\top \dot{S} S^\top S = -S^\top \dot{S} \tag{31}$$

And $\Omega = S^\top \dot{S} = -\Omega^\top$, confirming $\Omega \in \mathfrak{so}(d)$.

For the second claim, suppose $\dot{\widetilde{X}} = \widetilde{N}\widetilde{X}$ with $\widetilde{N} = \widetilde{N}^\top$. Then $\widetilde{X}^\top \dot{\widetilde{X}} = \widetilde{X}^\top \widetilde{N}\widetilde{X}$ is symmetric (as $\widetilde{N}$ is symmetric and $\widetilde{X}^\top \widetilde{N}\widetilde{X}$ is a congruence).

But from our expression:

$$\widetilde{X}^\top \dot{\widetilde{X}} = \widetilde{X}^\top N \widetilde{X} + \widetilde{X}^\top \widetilde{X}\Omega \tag{32}$$

The first term $\widetilde{X}^\top N \widetilde{X}$ is symmetric (since N is). The second term $\left(\widetilde{X}^\top \widetilde{X}\right)\Omega$ is the product of a symmetric positive-definite matrix and a skew-symmetric matrix.

If B is symmetric positive-definite and Ω is skew-symmetric, then $(B\Omega)^\top = \Omega^\top B = -\Omega B$. For $B\Omega$ to be symmetric, we need $B\Omega = -\Omega B$, i.e., $B\Omega + \Omega B = 0$. Since B is positive-definite, this holds only if $\Omega = 0$.

Thus $\widetilde{X}^\top \dot{\widetilde{X}}$ symmetric implies $\Omega = 0$, i.e., S is constant (no gauge drift). □

Corollary 1 The constraint that learned dynamics have the form $\dot{X} = NX$ with N symmetric automatically enforces horizontal alignment (no gauge velocity contamination).

This theorem explains the mechanism: random ASE gauge artifacts $R^{(t)}$ introduce skew-symmetric contamination $\widetilde{X}\Omega$ that cannot be absorbed into symmetric N. The joint optimization, by requiring symmetric dynamics, implicitly selects $Q_t \approx \left(R^{(t)}\right)^{-1}$—exactly the gauges that remove the ASE artifacts.

2.1.10.3 Extension to gauge-invariant dynamics

The analysis extends beyond linear dynamics. Consider any dynamics of the form:

$$\dot{X} = N(P)X \quad \text{where } N(P) = N(P)^\top \tag{33}$$

with N depending on $P = XX^\top$ (which is gauge-invariant: $\tilde{P} = \widetilde{X}\widetilde{X}^\top = XSS^\top X^\top = XX^\top = P$).

Proposition 5 For dynamics $\dot{X} = N(P)X$ with $N(P)$ symmetric, the gauge velocity contamination analysis applies unchanged: wrong gauges produce apparent dynamics $\dot{\widetilde{X}} = N(P)\widetilde{X} + \widetilde{X}\Omega$ with skew Ω, which cannot be fit by symmetric dynamics.

Examples of gauge-invariant symmetric dynamics:

1. **Polynomial:** $N(P) = \sum_{k=0}^{K} \alpha_k P^k$ (symmetric since P is)
2. **Spectral:** $N(P) = \sum_{i=1}^{n} \varphi(\lambda_i)\Pi_i$ where $P = \sum_i \lambda_i \Pi_i$ is the spectral decomposition
3. **Heat kernel:** $N(P) = e^{\beta P} - I$ (matrix exponential of symmetric is symmetric)
4. **Graph Laplacian:** $N(P) = D - P$ where $D = \mathrm{diag}(P\mathbf{1})$ is the degree matrix

All of these automatically enforce horizontal alignment when used as the dynamics family $\mathcal{F}$.

Proposition 6 (Identifiability under Generic Dynamics). Suppose the true dynamics M^* has distinct eigenvalues and the true trajectory $\{X^{(t)}\}$ spans $\mathbb{R}^n$ (i.e., $[X^{(0)} \mid \cdots \mid X^{(T-1)}]$ has full row rank). Then the gauges $\{Q_t\}$ are identifiable up to a global $O(d)$ transformation: any minimizer of Equation 22 satisfies $Q_t = Q^* R_t^*$ for some fixed $Q^* \in O(d)$.

Proof. (Sketch) If M is symmetric with distinct eigenvalues, its eigenspaces are one-dimensional. The constraint $\widehat{X}^{(t+1)} R_t^\top = M \widehat{X}^{(t)}$ for all t, combined with the trajectory spanning $\mathbb{R}^n$, determines M uniquely. Given M, each R_t is determined by the Procrustes problem (unique for generic data). The only remaining freedom is the choice of Q_0, which propagates as a global transformation. □

2.1.10.4 Extensions and limitations

Polynomial dynamics. The linear case extends to polynomial dynamics $\dot{X} = N(P)X$ where $N(P) = \sum_{k=0}^{K} \alpha_k P^k$. Since $P = P^\top$, we have $N(P) = N(P)^\top$, ensuring horizontal dynamics.

The discrete formulation becomes:

$$X^{(t+1)} \approx M\big(P^{(t)}\big) X^{(t)}, \quad M(P) = I + \delta t \sum_{k=0}^{K} \alpha_k P^k \tag{34}$$

The key observation is that $P^{(t)} = \widehat{X}^{(t)} Q_t Q_t^\top \widehat{X}^{(t)\top} = \widehat{X}^{(t)} \widehat{X}^{(t)\top}$ is **gauge-invariant**. Thus we can compute $P^{(t)}$ directly from the (unaligned) embeddings.

The optimization becomes:

$$\min_{\{R_t\},\{\alpha_k\}} \sum_{t=0}^{T-1} \|\widehat{X}^{(t+1)} R_t^\top - M\big(P^{(t)}\big) \widehat{X}^{(t)}\|_F^2 \tag{35}$$

Alternating optimization for polynomial case:

- ***α*-step (fix $\{R_t\}$):** Linear regression. With $Y_t = \widehat{X}^{(t+1)} R_t^\top - \widehat{X}^{(t)}$ and features $\Phi_t^{(k)} = \big(P^{(t)}\big)^k \widehat{X}^{(t)}$, solve:

  $$\min_{\{\alpha_k\}} \sum_t \|Y_t - \delta t \sum_k \alpha_k \Phi_t^{(k)}\|_F^2 \tag{36}$$

  This is ordinary least squares in the coefficients $\{\alpha_k\}$.

  Closed-form solution: Define Gram matrix $G \in \mathbb{R}^{(K+1)\times(K+1)}$ and target vector $b \in \mathbb{R}^{K+1}$:

  $$G_{kj} = \sum_t \langle \Phi_t^{(k)}, \Phi_t^{(j)} \rangle_F, \quad b_k = \sum_t \langle \Phi_t^{(k)}, Y_t \rangle_F \tag{37}$$

  Then $\alpha = G^{-1} b / \delta t$.

- ***R*-step (fix $\{\alpha_k\}$):** Same Procrustes subproblems as the linear case. For each t: compute $M\big(P^{(t)}\big) = I + \delta t \sum_k \alpha_k \big(P^{(t)}\big)^k$, then SVD of $\widehat{X}^{(t)\top} M\big(P^{(t)}\big)^\top \widehat{X}^{(t+1)} = U \Sigma V^\top$, and set $R_t = V U^\top$.

```
Input: Embeddings $\left\{\widehat{X}^{(t)}\right\}_{t=0}^{T}$, polynomial degree $K$, tolerance $\varepsilon$
Output: Aligned embeddings $\left\{\widetilde{X}^{(t)}\right\}$, coefficients $\{\alpha_k\}_{k=0}^{K}$
1. Precompute: $P^{(t)} \leftarrow \widehat{X}^{(t)}\widehat{X}^{(t)\top}$ for all $t$ (gauge-invariant!)
2. Initialize: $R_t^{(0)} \leftarrow \text{Procrustes}\left(\widehat{X}^{(t)}, \widehat{X}^{(t+1)}\right)$ for all $t$
3. Repeat until $|\mathcal{L}^{(k)} - \mathcal{L}^{(k-1)}| < \varepsilon$:
    • $\alpha$-step:
        ‣ $Y_t \leftarrow \widehat{X}^{(t+1)}\left(R_t^{(k)}\right)^{\top} - \widehat{X}^{(t)}$ for all $t$
        ‣ $\Phi_t^{(j)} \leftarrow \left(P^{(t)}\right)^{j}\widehat{X}^{(t)}$ for all $t, j$
        ‣ $G_{kj} \leftarrow \sum_t \operatorname{tr}\left(\Phi_t^{(k)\top}\Phi_t^{(j)}\right)$; $b_k \leftarrow \sum_t \operatorname{tr}\left(\Phi_t^{(k)\top}Y_t\right)$
        ‣ $\alpha^{(k+1)} \leftarrow G^{-1}b/\delta t$
    • $R$-step: For each $t$:
        ‣ $M \leftarrow I + \delta t \sum_j \alpha_j^{(k+1)}\left(P^{(t)}\right)^{j}$
        ‣ SVD: $\widehat{X}^{(t)\top}M^{\top}\widehat{X}^{(t+1)} = U\Sigma V^{\top}$
        ‣ $R_t^{(k+1)} \leftarrow VU^{\top}$
4. Recover absolute gauges: $Q_0 \leftarrow I$, $Q_{t+1} \leftarrow Q_t R_t$ for $t = 0, ..., T-1$
5. Return: $\widetilde{X}^{(t)} \leftarrow \widehat{X}^{(t)}Q_t$, $\{\alpha_k\}$
```

Algorithm 2: Structure-constrained alignment for polynomial dynamics

Identifiability analysis for polynomial dynamics. A critical question: does the polynomial constraint actually identify the correct gauges, or might there be spurious solutions?

Since $P^{(t)}$ is gauge-invariant, choosing $\{\alpha_k\}$ completely determines $M\left(P^{(t)}\right) = I + \delta t \sum_k \alpha_k \left(P^{(t)}\right)^k$ for each t. The R-step then asks: does there exist $R_t \in O(d)$ such that $\widehat{X}^{(t+1)}R_t^{\top} = M\left(P^{(t)}\right)\widehat{X}^{(t)}$?

For the **true** coefficients $\{\alpha_k^*\}$, this equation holds exactly (with R_t undoing the ASE gauge jumps). For **wrong** coefficients, $M\left(P^{(t)}\right)\widehat{X}^{(t)}$ is generically **not** a rotation of $\widehat{X}^{((t+1)\}}$ in the d-dimensional column space, so Procrustes yields nonzero residual.

Why the constraint helps: The requirement that the **same** $\{\alpha_k\}$ must work for **all** time steps is restrictive when $P^{(t)}$ varies. Different $P^{(t)}$ produce different $M\left(P^{((t)\}}\right)$, each imposing constraints on the coefficients.

Proposition 7 (Coefficient Identifiability from Varying P). Suppose the matrices $\left\{\left(P^{(0)}\right)^k, \left(P^{(1)}\right)^k, ..., \left(P^{(T-1)}\right)^k\right\}_{k=0}^{K}$ are such that the system of equations

$$\sum_{k=0}^{K} \alpha_k \left(P^{(t)}\right)^k \widehat{X}^{(t)} = \widehat{X}^{(t+1)}R_t^{\top} \quad \text{for } t = 0, ..., T-1 \tag{38}$$

is overdetermined in $\{\alpha_k\}$ when the $\{R_t\}$ are fixed. Then for generic data, the coefficients $\{\alpha_k\}$ are determined (given correct gauges), and wrong gauges produce coefficient estimates inconsistent across time.

Failure modes:

1. **Slowly varying $\boldsymbol{P^{(t)}}$:** If $P^{(t)} \approx P$ for all t, the matrices $\left(P^{(t)}\right)^k$ are approximately constant. Different polynomial coefficients give similar $M(P)$, and the system becomes underdetermined. *Remedy:* Use longer time series with larger spacing, or use trajectories with significant P variation.
2. **Short time series:** With $T \leq K+1$ time transitions, the polynomial coefficients are underdetermined even with varying P. *Remedy:* Require $T \gg K+1$.
3. **Low embedding dimension:** For $d=1$, gauges are signs (± 1), giving 2^T discrete possibilities. The polynomial constraint may not distinguish all sign sequences. *Remedy:* Use higher-order constraints or cross-validation.
4. **Degenerate configurations:** If the trajectory $\widehat{X}^{(t)}$ lies in a low-dimensional subspace, Procrustes may have non-unique solutions. *Remedy:* Verify rank conditions; use regularization.

The core mechanism (restated): ASE introduces gauge errors $R^{(t)} \in O(d)$ that are essentially "random" (determined by eigendecomposition conventions, not dynamics). The true trajectory $X^{(t)}$ satisfies $X^{(t+1)} = M^*\left(P^{(t)}\right)X^{(t)}$. In scrambled coordinates: $\widehat{X}^{(t+1)} = M^*\left(P^{(t)}\right)\widehat{X}^{(t)}S^{(t)}$ where $S^{(t)} = R^{(t)}\left(R^{(t+1)}\right)^{-1}$.

The joint optimization seeks $\{R_t\}$ undoing the $S^{(t)}$ and $\{\alpha_k\}$ matching the true M^*. For wrong $\{\alpha_k\}$, the Procrustes residuals are generically nonzero. The alternating algorithm descends toward solutions with small residual, which (under the conditions above) correspond to correct gauges and coefficients.

Condition	Risk	Mitigation
$P^{(t)}$ nearly constant	High	Larger δt; longer T
$T \leq K+1$	High	Use lower-degree polynomial or more data
$d = 1$	Medium	Cross-validate; multiple restarts
Noise $\gg$ signal	Medium	Larger n; regularization
Local minima	Medium	Good initialization; multiple restarts

Table 1: Identifiability risks and mitigations for polynomial dynamics.

Message-passing dynamics. Message-passing dynamics have the form:

$$\dot{x}_i = \sum_j P_{ij} g\left(x_i, x_j\right) \tag{39}$$

where $g : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ specifies how node j influences node i. This is a strong structural constraint: node i's velocity depends only on local information $\left(x_i, \left\{x_j : P_{ij} > 0\right\}\right)$.

Important special cases that yield horizontal dynamics:

1. **Neighbor attraction:** $g\left(x_i, x_j\right) = x_j$

$$\dot{x}_i = \sum_j P_{ij} x_j = (PX)_i \tag{40}$$

   In matrix form: $\dot{X} = PX$, which is $N = P$ (symmetric since $P = P^\top$). ✓

2. **Laplacian diffusion:** $g(x_i, x_j) = x_j - x_i$
$$\dot{x}_i = \sum_j P_{ij}(x_j - x_i) = (PX)_i - d_i x_i \tag{41}$$
where $d_i = \sum_j P_{ij}$. In matrix form: $\dot{X} = (P - D)X = -LX$ where $L = D - P$ is the graph Laplacian. Since $L = L^\top$, this is horizontal. ✓

3. **Heat kernel:** $g(x_i, x_j) = (e^{\beta P} - I)_{ij} x_j / P_{ij}$ (when $P_{ij} > 0$) Gives $\dot{X} = (e^{\beta P} - I)X$, symmetric for $\beta \in \mathbb{R}$. ✓

General linear message-passing. Consider $g(x_i, x_j) = W x_j$ for some $W \in \mathbb{R}^{d \times d}$:
$$\dot{X} = PXW \tag{42}$$
This mixes rows (via P) and columns (via W). For this to be horizontal, we need the induced N to be symmetric.

If $W = wI$ for scalar w, then $\dot{X} = wPX$, which is $N = wP$ (symmetric). ✓

For general W, the dynamics $\dot{X} = PXW$ are **not** of the form $\dot{X} = NX$ with N acting on rows only. However, if we vectorize: $\mathrm{vec}(\dot{X}) = (W^\top \otimes P)\ \mathrm{vec}(X)$, and $(W^\top \otimes P)$ is symmetric iff both W and P are symmetric.

The structural constraint for gauge identification. Message-passing enforces that velocity at node i is determined by **local** information. Random ASE gauge errors $R^{(t)}$ introduce **global** correlations that violate this structure.

Proposition 8 (Gauge Contamination in Message-Passing). Let $X(t)$ follow message-passing dynamics $\dot{x}_i = \sum_j P_{ij} g(x_i, x_j)$. Let $\widetilde{X} = XS$ for time-varying gauge $S(t) \in O(d)$. Then the apparent dynamics in the $\widetilde{X}$ frame are:
$$\dot{\tilde{x}}_i = \sum_j P_{ij} g(\tilde{x}_i S^{-1}, \tilde{x}_j S^{-1}) S + \tilde{x}_i \Omega \tag{43}$$
where $\Omega = S^{-1}\dot{S} \in \mathfrak{so}(d)$.

The second term $\tilde{x}_i \Omega$ is a **global** contribution (same Ω for all nodes) that does not factor through the message-passing structure.

This means: with wrong gauges, the "apparent velocities" include a global skew-symmetric component that cannot be explained by any local message-passing rule.

Joint optimization for parameterized message-passing. When $g = g_\theta$ is parameterized (e.g., a small neural network), we optimize:
$$\min_{\{Q_t\}, \theta} \sum_{t=0}^{T-1} \sum_{i=1}^{n} \| \tilde{x}_i^{(t+1)} - \tilde{x}_i^{(t)} - \delta t \sum_j P_{ij}^{(t)} g_{\theta\left(\tilde{x}_i^{(t)}, \tilde{x}_j^{(t)}\right)} \|^2 \tag{44}$$
where $\tilde{x}_i^{(t)} = \left(\widehat{X}^{(t)} Q_t\right)_{i\cdot}$.

Algorithm for message-passing dynamics:

Unlike the polynomial case, there's no closed-form solution for the θ-step. We use gradient-based optimization:

```
Input: Embeddings $\left\{\widehat{X}^{(t)}\right\}_{t=0}^{T}$, message-passing architecture $g_\theta$
Output: Aligned embeddings $\left\{\widetilde{X}^{(t)}\right\}$, learned parameters $\theta$
1. Initialize: $Q_t^{(0)} \leftarrow \prod_{s=0}^{t-1} \text{Procrustes}\left(\widehat{X}^{(s)}, \widehat{X}^{(s+1)}\right)$ (chained pairwise) $\theta^{(0)} \leftarrow$ random initialization
2. Repeat until convergence:
   • θ-step: Gradient descent on $\theta$ with $\{Q_t\}$ fixed:
         $\theta^{(k+1)} \leftarrow \theta^{(k)} - \eta \nabla_\theta \mathcal{L}\left(\left\{Q_t^{(k)}\right\}, \theta^{(k)}\right)$      (45)
   • Q-step: For each $t$, optimize $Q_t$ on $O(d)$ manifold:
         $Q_t^{(k+1)} \leftarrow \arg\min_{Q \in O(d)} \mathcal{L}_{t\left(Q;\theta^{(k+1)},\left\{Q_{s\neq t}^{(k)}\right\}\right)}$      (46)
     (via Riemannian gradient descent or Procrustes-like update)
3. Return: $\widetilde{X}^{(t)} \leftarrow \widehat{X}^{(t)} Q_t$, $\theta$
```

Algorithm 3: Structure-constrained alignment for message-passing dynamics

Why message-passing constrains gauges (intuition):

Consider two scenarios:

- **Correct gauges:** $\widetilde{X}^{(t)} \approx X^{(t)}$. The velocity $\tilde{x}_i^{(t+1)} - \tilde{x}_i^{(t)}$ truly depends only on local neighbors. The message-passing model fits well.
- **Wrong gauges:** The gauge error $S^{(t)} = R^{((t)\}} Q_t$ varies with t. The apparent velocity includes $\tilde{x}_i \Omega^{(t)}$ which is the **same** for all nodes (given their current position). This global coherence cannot be captured by local message-passing.

The message-passing loss will be higher for wrong gauges because it cannot fit the global gauge-velocity component.

Horizontal message-passing as a special case. When $g(x_i, x_j) = g(x_j, x_i)$ (symmetric interaction) and the aggregation is linear:

$$\dot{x}_i = \sum_j P_{ij} h(x_j) \tag{47}$$

for some $h : \mathbb{R}^d \to \mathbb{R}^d$, we can write $\dot{X} = PH(X)$ where $H(X)$ applies h row-wise.

For $h(x) = Wx$ with $W = W^\top$, we get $\dot{X} = PXW$, and if additionally $W = I$, this reduces to $\dot{X} = PX$—the polynomial case with $\alpha_0 = 0, \alpha_1 = 1$.

> *Remark.* The polynomial case $N(P) = \sum_k \alpha_k P^k$ is a **subset** of message-passing dynamics. Specifically, $\left(P^k X\right)_i = \sum_j \left(P^k\right)_{ij} x_j$ sums over k-hop neighbors. Message-passing with general g_θ can capture **nonlinear** local interactions that polynomials cannot.

2.1.10.5 Summary: Three dynamics families

Family	Form	Parameters	Algorithm
Linear	$\dot{X} = NX$, $N = N^\top$	$\frac{n(n+1)}{2}$ entries of N	Closed-form alternating (Algorithm 1)
Polynomial	$\dot{X} = \left(\sum_k \alpha_k P^k\right) X$	$K + 1$ coefficients	Closed-form alternating (Algorithm 2)
Message-passing	$\dot{x}_i = \sum_j P_{ij} g_{\theta(x_i, x_j)}$	$\lvert\theta\rvert$ (neural net)	Gradient-based (Algorithm 3)

Table 2: Comparison of dynamics families for structure-constrained alignment.

Key tradeoffs:

- **Linear:** Most general symmetric dynamics, but $O(n^2)$ parameters. Best when dynamics don't factor through P.
- **Polynomial:** Parsimonious ($K + 1$ parameters), closed-form updates, gauge-invariant features. Best when dynamics depend on network structure through powers of P.
- **Message-passing:** Most expressive (nonlinear local interactions), but requires gradient-based optimization and risks overfitting. Best for complex local dynamics.

Gauge identification strength:

All three families enforce **symmetric** or **local** structure that random ASE gauges violate.

- Linear/polynomial: Wrong gauges produce asymmetric N or inconsistent α_k across time.
- Message-passing: Wrong gauges produce global velocity correlations that local g cannot fit.

The polynomial family offers the best balance: parsimonious, closed-form, and leverages the gauge-invariance of P to separate the alignment problem from coefficient estimation.

Convergence. The alternating optimization decreases the objective at each step (both subproblems have closed-form solutions that achieve the global optimum given the other variables fixed). Thus the algorithm converges to a stationary point. However, the overall problem is non-convex, so we cannot guarantee convergence to the global optimum.

Proposition 9 (Monotonic Decrease). Let $\mathcal{L}^{(k)}$ denote the objective value after iteration k of Algorithm 1. Then $\mathcal{L}^{(k+1)} \leq \mathcal{L}^{(k)}$, with equality if and only if $\left(M^{(k)}, \left\{R_t^{(k)}\right\}\right)$ is a stationary point.

Good initialization is crucial. Starting from pairwise Procrustes alignments (ignoring the symmetry constraint) provides a reasonable warm start. Multiple random restarts can help escape poor local minima.

Limitations.

1. **Model mismatch:** If true dynamics lie outside $\mathcal{F}$, the method may find spurious alignments or fail to converge.
2. **Local minima:** The alternating optimization is non-convex; good initialization (e.g., from pairwise Procrustes) is important.

3. **Sample complexity:** With few time points or high noise, the constraint may be insufficient to determine gauges.
4. **Computational cost:** Each iteration requires T SVDs of $d \times d$ matrices (cheap) and one $n \times n$ linear solve (more expensive for large n).

> *Remark.* This approach inverts the usual pipeline: rather than "align then learn," we "learn to align." The dynamics model provides the inductive bias needed to resolve gauge ambiguity. This is philosophically similar to how physical constraints (symmetries, conservation laws) help identify coordinates in physics.

2.1.10.6 Effect of noise

In practice, we observe adjacency matrices $A^{(t)}$ from which we estimate $\widehat{X}^{(t)}$ via ASE. These estimates have error: $\widehat{X}^{(t)} = X^{(t)} R^{(t)} + E^{(t)}$ where $R^{(t)} \in O(d)$ is the gauge ambiguity and $E^{(t)}$ is statistical noise.

Under standard RDPG asymptotics [8], the rows of $E^{(t)}$ satisfy:

$$\|E_{i\cdot}^{(t)}\| = O_{p(1/\sqrt{n})} \tag{48}$$

with the noise being approximately Gaussian for large n.

Impact on structure-constrained alignment:

The objective becomes:

$$\mathcal{L} = \sum_t \left\| \left(X^{(t+1)} R^{(t+1)} + E^{(t+1)}\right) Q_{t+1} - M\left(X^{(t)} R^{(t)} + E^{(t)}\right) Q_t \right\|_F^2 \tag{49}$$

At the true solution ($Q_t = \left(R^{(t)}\right)^{-1}$ and $M = M^*$), the residual is:

$$\mathcal{L}^* = \sum_t \left\| E^{(t+1)} \left(R^{(t+1)}\right)^{-1} - M^* E^{(t)} \left(R^{(t)}\right)^{-1} \right\|_F^2 = O_{p(n/n)} = O_{p(1)} \tag{50}$$

The noise contributes $O(1/\sqrt{n})$ per entry, summed over n nodes, giving $O(1)$ total.

Key insight: The structure constraint helps **even with noise** because:
1. Noise is isotropic (no preferred gauge direction)
2. The symmetry constraint on M averages over many node pairs
3. Wrong gauges produce **systematic** asymmetry in M, distinguishable from random noise

However, for small n or large noise, the signal (symmetry violation from wrong gauge) may be overwhelmed by noise. A rough requirement is that the gauge-induced asymmetry exceeds noise:

$$\|M^* - M^{\text{wrong}}\|_F \gg \|E\|_F \;/\; \|X\|_F \approx 1/\sqrt{n} \tag{51}$$

2.2 Dynamics in Latent Space and Gauge Freedom

We now consider how latent positions evolve over time and what this implies for observable network structure.

2.2.1 The dynamical framework

We model the evolution of latent positions as a dynamical system:

$$\dot{X} = f(X) \tag{52}$$

where $X \in \left(B_+^d\right)^n$ collects all node positions and $f : \left(B_+^d\right)^n \to \mathbb{R}^{n\times d}$ specifies each node's velocity as a function of the current configuration.

Concretely, $\dot{X}_i = f(X)_i$ describes how node i's position $\vec{x}_i$ changes based on the positions of all nodes. For instance, f might encode attraction toward similar nodes or repulsion from dissimilar ones.

Since we observe networks (and hence probability matrices $P = XX^\top$), not latent positions directly, we need to understand what dynamics on X imply for P. By the product rule:

$$\dot{P} = \dot{X}X^\top + X\dot{X}^\top = f(X)X^\top + Xf(X)^\top \tag{53}$$

This is the **induced dynamics** on the observable. Any vector field f on latent space induces a corresponding evolution of the probability matrix.

2.2.2 Gauge freedom

The latent positions X are not uniquely determined by P. For any orthogonal matrix $Q \in O(d)$:

$$(XQ)(XQ)^\top = XQQ^\top X^\top = XX^\top = P \tag{54}$$

Thus X and XQ represent the *same* probability matrix P. This is the **gauge freedom** of RDPG: the equivalence class $[X] = \{XQ : Q \in O(d)\}$ corresponds to a single observable network structure.

Geometrically, gauge freedom means that a global rotation of all positions in latent space leaves all connection probabilities unchanged. This makes sense: dot products depend only on angles and magnitudes, which are preserved under orthogonal transformations.

This gauge freedom has profound implications for learning dynamics. If two configurations X and XQ are observationally indistinguishable, then any dynamics that moves along this equivalence class—rotating all positions by a common orthogonal transformation—produces no observable change in the network. We call such dynamics **invisible**.

> **Definition 7** (Observable vs. Invisible Dynamics). A vector field $f : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$ produces *observable dynamics* if $\dot{P} \neq 0$, where $\dot{P} = f(X)X^\top + Xf(X)^\top$. Otherwise, the dynamics are *invisible*—the parameterization changes but the graph structure is static.

Theorem 4 (Characterization of Invisible Dynamics). Let $X \in \mathbb{R}^{n\times d}$ have full column rank. A vector field f produces invisible dynamics if and only if $f(X) = XA$ for some skew-symmetric matrix $A \in \mathfrak{so}(d)$, i.e., $A^\top = -A$.

> *Remark.* The theorem is stated for $X \in \mathbb{R}^{n\times d}$ to emphasize that the gauge-theoretic structure holds in full generality. In practice, for $P = XX^\top$ to be a valid probability matrix, positions must satisfy the **inner product distribution** constraint [8]: for all pairs i, j, we require $\vec{x}_i \cdot \vec{x}_j \in [0, 1]$. A sufficient condition is $X \in \left(B_+^d\right)^n$, but other configurations (e.g., positions on a Hardy-Weinberg curve) also yield valid probabilities.

Proof. ($\Leftarrow$) Suppose $f(X) = XA$ with $A^\top = -A$. Then:

$$\dot{P} = XAX^\top + X(XA)^\top = XAX^\top + XA^\top X^\top = X(A + A^\top)X^\top = 0 \tag{55}$$

($\Rightarrow$) Suppose $\dot{P} = f(X)X^\top + Xf(X)^\top = 0$.

Decompose $f(X) = XA + W$ where $A = \left(X^\top X\right)^{-1} X^\top f(X)$ and $X^\top W = 0$ (i.e., W lies in the orthogonal complement of $\mathrm{col}(X)$).

Substituting into $\dot{P} = 0$:

$$0 = (XA + W)X^\top + X(XA + W)^\top = X(A + A^\top)X^\top + WX^\top + XW^\top \tag{56}$$

For X with full column rank, consider the constraint $X^\top W = 0$ combined with the equation above. Taking the projection onto $\mathrm{col}(X)$: multiplying on left by $\left(X^\top X\right)^{-1} X^\top$ and on right by $X\left(X^\top X\right)^{-1}$:

$$0 = A + A^\top + \left(X^\top X\right)^{-1} X^\top W X^\top X \left(X^\top X\right)^{-1} + \text{similar term} \tag{57}$$

Since $X^\top W = 0$, the middle terms vanish, giving $A + A^\top = 0$.

For the remaining equation $WX^\top + XW^\top = 0$ with $X^\top W = 0$: this is a symmetric matrix equation. For generic full-rank X, the only solution is $W = 0$. □

Corollary 2 The space of invisible dynamics is isomorphic to $\mathfrak{so}(d)$, with dimension $\frac{d(d-1)}{2}$. For $d = 2$, this is 1-dimensional (a single rotation rate); for $d = 3$, it is 3-dimensional.

The invisible dynamics $\dot{X} = XA$ are infinitesimal rotations along gauge orbits. In particular, *uniform rotation around the origin* satisfies $\dot{X}_i = X_i A$, producing $\dot{P} = 0$—the embedding rotates but the network is static.

Crucially, other rotational dynamics *are* observable. Rotation around the *origin* is gauge (invisible), but circulation around a *nonzero centroid* is observable:

Proposition 10 (Centroid Circulation). Dynamics of the form $\dot{X}_i = (X_i - |(X))A$ with $|(X) \neq 0$ and $A \in \mathfrak{so}(d)$ produce observable changes in P.

Proof. Let $|(X) = \frac{1}{n}\sum_i X_i$ be the centroid. The dynamics $\dot{X}_i = (X_i - |(X))A$ can be rewritten as:

$$\dot{X}_i = X_i A - |(X)A \tag{58}$$

In matrix form with $\mathbf{1} \in \mathbb{R}^n$ the all-ones vector:

$$\dot{X} = XA - \mathbf{1}|(X)^\top A \tag{59}$$

Computing $\dot{P}$:

$$\begin{aligned}\dot{P} &= \dot{X}X^\top + X\dot{X}^\top \\ &= (XA - \mathbf{1}|(X)^\top A)X^\top + X\left(XA - \mathbf{1}|(X)^\top A\right)^\top \\ &= XAX^\top + XA^\top X^\top - \mathbf{1}|(X)^\top AX^\top - XA^\top\ |(X)\mathbf{1}^\top\end{aligned} \tag{60}$$

The first two terms cancel since $A + A^\top = 0$. Let $v = A^\top\ |(X) = -A\ |(X)$:

$$\dot{P} = \mathbf{1}v^\top X^\top + Xv\mathbf{1}^\top \tag{61}$$

Entry-wise: $\dot{P}_{ij} = v \cdot X_j + X_i \cdot v$.

This vanishes for all i, j only if $v = 0$, i.e., $A\ |(X) = 0$. For generic $|(X) \neq 0$ and $A \neq 0$, we have $\dot{P} \neq 0$. □

Interpretation. Circulation around the centroid decomposes as:

$$\dot{X}_i = \underbrace{X_i A}_{\text{invisible (gauge)}} - \underbrace{|(X)A}_{\text{shared drift (observable)}} \tag{62}$$

The first term is pure gauge. The second is a constant velocity applied to all nodes, which shifts all dot products and hence changes P.

Similarly, differential rotation rates are observable:

Proposition 11 (Differential Rotation is Observable). If nodes have different rotation rates:

$$\dot{X}_i = X_i A_i, \quad A_i \in \mathfrak{so}(d) \tag{63}$$

then:

$$\dot{P}_{ij} = X_i(A_i - A_j)X_j^\top \tag{64}$$

This is generically nonzero when $A_i \neq A_j$.

Proof.

$$\begin{aligned} \dot{P}_{ij} &= \dot{X}_i X_j^\top + X_i \dot{X}_j^\top = (X_i A_i)X_j^\top + X_i\big(X_j A_j\big)^\top \\ &= X_i A_i X_j^\top + X_i A_j^\top X_j^\top = X_i A_i X_j^\top - X_i A_j X_j^\top \\ &= X_i\big(A_i - A_j\big)X_j^\top \end{aligned} \tag{65}$$

using $A_j^\top = -A_j$. This is nonzero when $A_i \neq A_j$ and X_i, X_j are generic. □

Dynamics	$\dot{P} = 0$?	**Observable?**
$\dot{X} = XA$ (uniform rotation around origin)	Yes	No
$\dot{X}_i = (X_i - |(X))A$ with $|(X) \neq 0$	No	Yes
$\dot{X}_i = X_i A_i$ with $A_i \neq A_j$	No	Yes
$\dot{X}_i = \alpha X_i$ (radial scaling)	No	Yes
$\dot{X}_i = \sum_j w_{ij}\big(X_j - X_i\big)$ (attraction/repulsion)	No	Yes

Table 3: Classification of dynamics by observability.

A natural question arises: given that some dynamics are invisible, can we still learn the observable part?

Theorem 5 (Identifiability Modulo Gauge). Let $X \in \mathbb{R}^{n\times d}$ have full column rank. Given $\dot{P}$ and X, the vector field $f(X)$ is uniquely determined up to gauge:

$$f(X) = F + XA \tag{66}$$

where F is any solution to $\dot{P} = FX^\top + XF^\top$ and $A \in \mathfrak{so}(d)$ is arbitrary.

Proof. Consider the linear map $\mathcal{L} : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times n}$ defined by $\mathcal{L}(F) = FX^\top + XF^\top$. The equation $\dot{P} = \mathcal{L}(f(X))$ determines $f(X)$ up to elements of $\ker(\mathcal{L})$.

By Theorem 4, $F \in \ker(\mathcal{L})$ if and only if $F = XA$ for some skew-symmetric $A \in \mathfrak{so}(d)$. Thus if F_1 and F_2 both satisfy $\dot{P} = F_i X^\top + XF_i^\top$, then $F_1 - F_2 \in \ker(\mathcal{L})$, so $F_1 = F_2 + XA$ for some $A \in \mathfrak{so}(d)$.

To show existence: the image of $\mathcal{L}$ consists of symmetric matrices in the row/column space of X. Since $\dot{P} = \dot{X}X^\top + X\dot{X}^\top$ is symmetric and $\dot{X}$ lies in this space (being a velocity of positions), $\dot{P} \in \mathrm{im}(\mathcal{L})$. □

Theorem 6 (Gauge Equivalence). Two vector fields f and $\tilde{f}$ are gauge equivalent (induce the same $\dot{P}$) if and only if:

$$f(X) - \tilde{f}(X) = XA(X) \tag{67}$$

for some $\mathfrak{so}(d)$-valued function $A(X)$.

Proof. Apply Theorem 4 to the difference $h = f - \tilde{f}$. □

Corollary 3 (Canonical Decomposition). Any vector field decomposes uniquely as:

$$f(X) = f_{\text{phys}}(X) + XA(X) \tag{68}$$

where f_{phys} determines $\dot{P}$ and XA is pure gauge.

This decomposition clarifies what can be learned: the "physical" content—what affects the observable—is uniquely determined; only the coordinate-dependent form varies with gauge choice.

Gauge-Free Decomposition of $\dot{X}$. Any velocity $\dot{X}$ decomposes uniquely as $\dot{X} = XA + W$ where $A = (X^\top X)^{-1} X^\top \dot{X} \in \mathbb{R}^{d\times d}$ and $W \perp \mathrm{col}(X)$. Further decomposing $A = A_{\text{sym}} + A_{\text{skew}}$:

Component	Contributes to $\dot{P}$?	Interpretation
XA_{sym}	Yes	Radial/stretching dynamics
XA_{skew}	No	Pure rotation (gauge)
W	Yes	Rotates $\mathrm{col}(P)$ in $\mathbb{R}^n$

Table 4: Decomposition of $\dot{X}$ into observable and gauge components.

The observable content of $\dot{X}$ is (A_{sym}, W).

Implications for learning. When we train on estimated positions $\widehat{X}(t)$, the Procrustes alignment fixes a consistent gauge. The learned f determines $\dot{P}$ correctly, but a different alignment procedure would yield a gauge-equivalent $f + XA$.

2.3 Neural ODE Dynamics

After embedding, we have a sequence of latent positions $\left\{\widehat{X}_t\right\}_{t=1}^{T}$. We flatten each $\widehat{X}_t \in \mathbb{R}^{n \times d}$ into a vector $\boldsymbol{u}_t \in \mathbb{R}^{nd}$ and model the dynamics as:

$$\frac{d\boldsymbol{u}}{dt} = f_{\theta(\boldsymbol{u})} \tag{69}$$

where f_θ is a neural network with parameters θ.

We parameterize f_θ as a fully-connected network with architecture:

$$f_\theta : \mathbb{R}^{nd} \xrightarrow{\text{Dense}} \mathbb{R}^{128} \xrightarrow{\text{celu}} \mathbb{R}^{128} \xrightarrow{\text{celu}} \mathbb{R}^{64} \xrightarrow{\text{celu}} \mathbb{R}^{nd} \tag{70}$$

Training minimizes the prediction error:

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \|\boldsymbol{u}_t - \hat{\boldsymbol{u}}_{t(\theta)}\|_2^2 + \lambda \mathcal{L}_{\text{prob}} \tag{71}$$

where $\hat{\boldsymbol{u}}_t$ is obtained by integrating the Neural ODE from $\boldsymbol{u}_1$, and $\mathcal{L}_{\text{prob}}$ penalizes predicted probabilities outside $[0, 1]$:

$$\mathcal{L}_{\text{prob}} = \sum_{i \neq j} \max\left(0, -P_{ij}\right) + \max\left(0, P_{ij} - 1\right) \tag{72}$$

We use a two-stage optimization: Adam for initial exploration followed by Lion for fine-tuning. Gradients are computed via adjoint sensitivity analysis for memory efficiency [9].

2.4 Universal Differential Equations

When domain knowledge suggests a particular functional form for the dynamics, we can incorporate it via Universal Differential Equations (UDEs) [15]. The vector field decomposes as:

$$f(\boldsymbol{u}) = f_{\text{known}}(\boldsymbol{u}; \varphi) + f_{\text{NN}}(\boldsymbol{u}; \theta) \tag{73}$$

where f_{known} encodes known physics with parameters φ, and f_{NN} is a neural network that learns residual corrections.

For RDPG dynamics, gauge theory (Section 2.2) suggests a particularly elegant form.

Theorem 7 (Equivariant Dynamics). Let $X \in \mathbb{R}^{n \times d}$ have full column rank. Any $O(d)$-equivariant vector field $f : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}$ has the form:

$$f(X) = N(P) \cdot X \tag{74}$$

where $N : \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$ depends only on $P = XX^\top$.

Proof. Define $N(X) \coloneqq f(X)X^\dagger$ where $X^\dagger = \left(X^\top X\right)^{-1} X^\top$ is the Moore-Penrose pseudoinverse.

For equivariance, we require $f(XQ) = f(X)Q$ for all $Q \in O(d)$. Then:

$$N(XQ) = f(XQ)(XQ)^\dagger = f(X)Q\big(Q^\top X^\dagger\big) = f(X)X^\dagger = N(X) \tag{75}$$

So N is constant on $O(d)$-orbits. Since orbits are indexed by $P = XX^\top$ (two matrices $X, \widetilde{X}$ lie on the same orbit iff $XX^\top = \widetilde{X}\widetilde{X}^\top$), we have $N = N(P)$. □

This form is automatically gauge-consistent since N depends on the observable P, not the gauge-dependent X. The key question is: how should we constrain N to eliminate gauge freedom?

Theorem 8 (Gauge Dynamics are Not Symmetric). The invisible (gauge) dynamics $\dot{X} = XA$ with $A \in \mathfrak{so}(d)$ correspond to $N = XAX^\dagger$. For generic full-rank X and nonzero A, this N is **not symmetric**.

Proof. For $X = I_d$ (taking $n = d$), we have $N = A$, which is skew-symmetric.

For general X with thin SVD $X = U\Sigma V^\top$, we get:

$$N = XAX^\dagger = U\Sigma V^\top AV\Sigma^{-1}U^\top = UBU^\top \tag{76}$$

where $B = \Sigma(V^\top AV)\Sigma^{-1}$.

Since $V^\top AV$ is skew-symmetric (as A is) and Σ generically has distinct singular values, the conjugation by Σ and Σ^{-1} breaks the skew-symmetry: $B \neq B^\top$ unless $A = 0$.

Thus $N = UBU^\top$ is neither symmetric nor skew-symmetric for generic X and nonzero A. □

This theorem is the key insight: *gauge directions correspond to non-symmetric* N. Therefore, constraining N to be symmetric eliminates gauge:

Theorem 9 (Gauge Elimination via Symmetry). Constraining $N(P) = N(P)^\top$ (symmetric) eliminates all non-trivial gauge freedom. Any symmetric N with $NX \neq 0$ produces observable dynamics ($\dot{P} \neq 0$). Moreover, symmetric N can produce *any* realizable $\dot{P}$—no expressivity is lost.

Proof. Suppose $N = N^\top$ and $\dot{P} = NP + PN = 0$.

Let $P = V\Lambda V^\top$ be the spectral decomposition with $\Lambda = \operatorname{diag}(\lambda_1, ..., \lambda_d, 0, ..., 0)$ and $\lambda_i > 0$ for $i \leq d$.

Define $\widetilde{N} = V^\top NV$ (symmetric since N is). The condition $NP + PN = 0$ transforms to:

$$\widetilde{N}\Lambda + \Lambda\widetilde{N} = 0 \tag{77}$$

Entry-wise: $\widetilde{N}_{ij}\big(\lambda_i + \lambda_j\big) = 0$.

Case analysis:

- For $i, j \leq d$: $\lambda_i + \lambda_j > 0$, so $\widetilde{N}_{ij} = 0$.
- For $i \leq d$, $j > d$: $\lambda_i + 0 = \lambda_i > 0$, so $\widetilde{N}_{ij} = 0$.
- For $i > d$, $j \leq d$: by symmetry $\widetilde{N}_{ij} = \widetilde{N}_{ji} = 0$.
- For $i, j > d$: the constraint $0 \cdot \widetilde{N}_{ij} = 0$ is vacuous.

Therefore $\widetilde{N} = \begin{pmatrix} 0 & 0 \\ 0 & \widetilde{N}_{22} \end{pmatrix}$ where $\widetilde{N}_{22} \in \mathbb{R}^{(n-d)\times(n-d)}$ is arbitrary symmetric (supported on $\operatorname{null}(P)$).

Since $\mathrm{col}(X) = \mathrm{col}(V_1)$ where V_1 comprises the first d columns of V, we can write $X = V_1 R$ for invertible R. Then:

$$NX = V\widetilde{N}V^\top V_1 R = V\widetilde{N}\begin{pmatrix} I_d \\ 0 \end{pmatrix} R = V\begin{pmatrix} 0 \\ 0 \end{pmatrix} = 0 \tag{78}$$

Contrapositive: If $N = N^\top$ and $NX \neq 0$, then $NP + PN \neq 0$, so $\dot{P} \neq 0$. □

This suggests a principled UDE architecture:

$$N(P) = N_{\text{known}}(P) + N_{\text{NN}}(P), \quad \text{both symmetric} \tag{79}$$

where N_{known} might be a polynomial in P (encoding local neighbor influence) and N_{NN} learns corrections.

We consider three parameterization classes in order of increasing expressivity:

Polynomial $N(P)$. The most parsimonious form:

$$N = \alpha_0 I + \alpha_1 P + \alpha_2 P^2 + \ldots + \alpha_k P^k \tag{80}$$

with only $k + 1$ learnable scalars. The interpretation is intuitive: $\alpha_0 I$ represents intrinsic node dynamics, $\alpha_1 P$ captures direct neighbor influence (one-hop interactions), and $\alpha_2 P^2$ captures two-hop effects through shared neighbors. For many network dynamics, degree $k \leq 2$ suffices.

Gauge invariance of learned parameters. A key advantage of expressing dynamics in terms of P rather than X directly: the scalar coefficients $\alpha_0, \alpha_1, \ldots$ are *gauge-invariant*. Since $P = XX^\top$ is unchanged by orthogonal transformations $X \mapsto XQ$, and $N(P)$ depends only on P, the learned α_k values are independent of the coordinate system chosen by SVD and Procrustes alignment. This means coefficients learned in *any* gauge (including the DUASE-estimated coordinates) can be applied to *any other* gauge (including the true positions)—the "learn anywhere, apply everywhere" principle. By contrast, dynamics expressed directly in X coordinates (e.g., $\dot{X}_1 = aX_1 + bX_2$) would have coefficients that depend on the specific basis chosen.

Pairwise kernel $N(P)$. A flexible homogeneous form:

$$N_{ij} = \begin{cases} \kappa(P_{ij}) i \neq j \\ h(P_{ii})\, i = j \end{cases} \tag{81}$$

where $\kappa, h : [0, 1] \to \mathbb{R}$ are learned functions (small neural networks or parametric forms). Symmetry is automatic since $P_{ij} = P_{ji}$. The kernel κ can capture nonlinear responses to connection probability, such as threshold effects or saturation.

General symmetric $N(P)$. A neural network that outputs the upper triangle of a symmetric matrix:

$$\mathrm{NN} : \mathrm{uptri}(P) \mapsto \mathrm{uptri}(N) \tag{82}$$

with $\frac{n(n+1)}{2}$ inputs and outputs. This is the most expressive but least parsimonious option.

Table 5 summarizes the parameter counts.

Architecture	Parameters	Expressivity
Polynomial ($k = 1$)	2	Low
Polynomial ($k = 2$)	3	Low
Pairwise kernel (16-16 NN)	≈300	Medium
General symmetric NN	≈5,000	High
Standard Neural ODE	≈10,000	Highest

Table 5: Parameter counts for N(P)X architectures vs. standard Neural ODE.

The dynamics $\dot{X} = NX$ have a clear physical interpretation: node i's velocity is $\dot{X}_i = \sum_j N_{ij} X_j$, a weighted combination of all positions where $N_{ij} > 0$ indicates attraction and $N_{ij} < 0$ indicates repulsion. When the true dynamics have this form, the polynomial parameterization can recover the exact coefficients with orders of magnitude fewer parameters than a generic neural network.

2.5 Realizable Dynamics and Model Diagnostics

Beyond gauge freedom, RDPG dynamics face a fundamental geometric constraint: the probability matrix $P = XX^\top$ lives on a low-dimensional manifold, so most symmetric perturbations $\dot{P}$ are not achievable.

Proposition 12 (Tangent Space Constraint). Let $V \in \mathbb{R}^{n\times d}$ be an orthonormal basis for $\mathrm{col}(P) \subset \mathbb{R}^n$, and $V_\perp \in \mathbb{R}^{n\times(n-d)}$ span its orthogonal complement. Any realizable $\dot{P}$ satisfies:

$$V_\perp^\top \dot{P}\, V_\perp = 0 \tag{83}$$

The realizable tangent space has dimension $nd - \frac{d(d-1)}{2}$.

Proof. Any realizable $\dot{P} = FX^\top + XF^\top$ for some $F \in \mathbb{R}^{n\times d}$. Since $\mathrm{col}(X) = \mathrm{col}(V)$, we have $X = VR$ for invertible $R \in \mathbb{R}^{d\times d}$. Then:

$$V_\perp^\top \dot{P}\, V_\perp = V_\perp^\top F R^\top V^\top V_\perp + V_\perp^\top V R F^\top V_\perp = 0 + 0 = 0 \tag{84}$$

using $V^\top V_\perp = 0$. Conversely, any symmetric $\dot{P}$ with $V_\perp^\top \dot{P} V_\perp = 0$ can be written in this form. □

Interpretation of blocks. Any symmetric $n \times n$ matrix M decomposes as:

$$M = \underbrace{VAV^\top}_{\text{range-range}} + \underbrace{VBV_\perp^\top + V_\perp B^\top V^\top}_{\text{range-null cross}} + \underbrace{V_\perp C V_\perp^\top}_{\text{null-null}} \tag{85}$$

For realizable $\dot{P}$: the A and B blocks can be arbitrary, but $C = 0$ always.

The null-null block represents "structure in the orthogonal complement"—dynamics that would increase the rank of P. If we fit an RDPG model and find systematic residuals with $C \neq 0$, this suggests the latent dimension d is too small or the RDPG model is inappropriate.

Corollary 4 (Dimension Count).

$$\dim(T_P \mathcal{M}_d) = nd - \frac{d(d-1)}{2} \tag{86}$$

This equals the dimension of X-space (nd) minus the gauge freedom ($\frac{d(d-1)}{2}$)—exactly the observable degrees of freedom.

Example. For $n = 10$ nodes and $d = 2$ dimensions: symmetric matrices have 55 degrees of freedom, but only 19 directions are realizable. The remaining 36 directions would require increasing the rank of P—that is, increasing the latent dimension d.

Model diagnostic. This constraint provides a principled diagnostic for model adequacy. If observed dynamics have structure in the "null-null" block $V_\perp^\top \dot{P} V_\perp$, this indicates one of two possibilities:

1. **Model misspecification**: The true dynamics do not preserve low-rank structure, so RDPG embedding is inappropriate.
2. **Dimensional emergence**: The latent dimension d is increasing over time—new factors are emerging in the network structure.

In practice, the constraint is automatically satisfied by dynamics of the form $\dot{X} = N(P)X$, since $\dot{P} = NP + PN$ has the required tangent structure by construction. Violations in fitted residuals suggest the chosen d may be too small.

2.6 Probability Constraints

For edge probabilities to be valid, we require $P_{ij} \in [0, 1]$ for all i, j. A sufficient condition is that all node positions lie in the positive orthant of the unit ball:

$$B_+^d = \{x \in \mathbb{R}^d : x \geq 0, \|x\| \leq 1\} \tag{87}$$

If $X_i \in B_+^d$ for all i, then $P_{ij} = X_i \cdot X_j \in [0, 1]$ by non-negativity of coordinates and Cauchy-Schwarz.

However, $\left(B_+^d\right)^n$ is *not* a fundamental domain: one cannot always rotate n vectors simultaneously into the positive orthant. For example, two orthogonal unit vectors in $\mathbb{R}^2$ cannot both have non-negative coordinates after any rotation.

We handle constraints via a barrier loss:

$$\mathcal{L}_{\text{prob}} = \gamma \sum_{i,j} \left[\max\left(0, -P_{ij}\right)^2 + \max\left(0, P_{ij} - 1\right)^2\right] \tag{88}$$

This encourages learned dynamics to remain in the valid region without breaking the ODE structure. Unlike projection-based approaches that clamp values, the barrier loss maintains differentiability throughout training.

Practical note. While the B_+^d constraint provides a convenient sufficient condition for mathematical analysis, our experiments show that explicit projection onto B_+^d is neither necessary nor beneficial for numerical learning. Euclidean ODE solvers with barrier losses maintain valid probabilities while preserving the natural geometry of the embedding space. Projecting onto B_+^d

can distort the geometry and introduce artifacts that complicate learning. The B_+^d framework remains valuable for theoretical analysis of constraint satisfaction, but practitioners should prefer unconstrained optimization with soft penalties.

2.7 Symbolic Regression

The trained Neural ODE provides accurate predictions but remains a black box. To extract interpretable dynamics, we apply symbolic regression to find closed-form expressions approximating f_θ.

Given the learned vector field $f_{\theta(\boldsymbol{u})}$, we seek symbolic expressions $g(\boldsymbol{u})$ from a grammar of operations (polynomials, trigonometric functions, etc.) that minimize:

$$\min_g \int \|f_{\theta(\boldsymbol{u})} - g(\boldsymbol{u})\|^2 \, \mathrm{d}\boldsymbol{u} + \text{complexity}(g) \tag{89}$$

The complexity penalty encourages parsimonious expressions. We use genetic programming [16] to search the space of symbolic expressions.

Gauge dependence of symbolic equations. A critical caveat: recovered equations are *gauge-dependent*. The symbolic form depends on the coordinate system fixed by SVD and Procrustes alignment. In a rotated basis, $\dot{X}_1 = \omega X_2$ might appear as $\dot{Y}_1 = aY_1 + bY_2$—different-looking equations producing identical $P(t)$.

What *is* gauge-invariant:

- Eigenvalues of the linearization (frequencies, decay rates)
- Equilibrium structure (existence, stability type)
- Qualitative behavior (oscillatory, stable, chaotic)

For truly coordinate-free equations, one could regress $\dot{P}_{ij}$ directly as functions of P, though at the cost of higher dimensionality.

3 Data: Synthetic Temporal Networks

We evaluate our framework on three synthetic temporal networks with known generating processes. This allows us to assess whether extracted equations match ground truth. Crucially, all three systems exhibit *observable* dynamics in the sense of Section 2.2 —the latent position changes produce actual changes in P.

3.1 Single Community Oscillation

A network of $n = 5$ nodes where connection probabilities oscillate sinusoidally. All nodes belong to a single community whose internal connectivity varies periodically. The ground-truth dynamics follow:

$$\frac{dX_{i,1}}{dt} = \omega X_{i,2}, \quad \frac{dX_{i,2}}{dt} = -\omega X_{i,1} \tag{90}$$

producing circular trajectories in embedding space. This is observable because nodes circulate around a nonzero centroid (Proposition 10), not the origin.

3.2 Two Communities Merging

A network of $n = 11$ nodes initially partitioned into two separate communities. Over time, the communities gradually merge into a single cohesive group. This models scenarios like organizational mergers or ecosystem succession. The dynamics involve attraction between nodes (Table 3), which is observable as it changes pairwise dot products.

3.3 Long-Tailed Degree Distribution

A network of $n = 36$ nodes with a power-law degree distribution. This tests whether our method handles networks with heterogeneous node connectivity, which are common in real-world systems due to preferential attachment.

4 Results

4.1 Training Performance

Table 6 summarizes the training results across all three systems.

Dataset	n	d	**Final MSE**
Single community oscillation	5	2	0.114
Two communities merging	11	2	1.169
Long-tailed distribution	36	2	0.159

Table 6: Training results for three synthetic temporal networks.

The single community and long-tail systems achieve low reconstruction error, while the merging communities system is more challenging. This may reflect the more complex dynamics of community reorganization.

4.2 Embedding Trajectories

Figure below (placeholder) shows example embedding trajectories comparing ground truth (from data) with Neural ODE predictions. The model successfully captures the qualitative dynamics in all cases.

4.3 Symbolic Regression

For the single community oscillation system, symbolic regression recovers equations of the form:

$$\frac{dX_1}{dt} \approx aX_2, \quad \frac{dX_2}{dt} \approx -aX_1 \tag{91}$$

matching the ground-truth harmonic oscillator dynamics. This demonstrates that our framework can recover interpretable, mechanistically meaningful equations from network observations alone.

4.4 Gauge-Consistent Architecture Comparison

To test whether the theoretically-motivated $\dot{X} = N(P)X$ form offers practical advantages, we compare three architectures on synthetic pairwise dynamics:

$$\dot{X} = (\alpha I + \beta P)X \tag{92}$$

with $\alpha = -0.02$ (slight contraction) and $\beta = 0.001$ (pairwise attraction). This dynamics has exactly the polynomial $N(P)X$ form with degree 1, providing a fair test where the correct inductive bias should help.

We compare:

1. **Standard Neural ODE**: Generic $f_{\theta(X)}$ with ≈10,000 parameters
2. **Polynomial $\boldsymbol{N(P)X}$**: $N = \alpha_0 I + \alpha_1 P$ with 2 parameters
3. **Kernel $\boldsymbol{N(P)X}$**: $N_{ij} = \kappa(P_{ij})$ with ≈300 parameters

Table 7 summarizes the results.

Architecture	**Parameters**	**MSE**	**Parameter Recovery**
Standard Neural ODE	≈10,000	—	N/A
Polynomial $N(P)X$	2	—	$\hat{\alpha}_0 \approx ?$, $\hat{\alpha}_1 \approx ?$
Kernel $N(P)X$	≈300	—	N/A

Table 7: Architecture comparison on pairwise dynamics ($n = 30$, $d = 2$).

The polynomial architecture offers a dramatic reduction in parameters (5000× fewer than standard NN) while potentially recovering the true dynamical coefficients. When the inductive bias matches the true dynamics, parsimony does not sacrifice accuracy.

5 Discussion

We presented a framework for learning interpretable dynamics of temporal networks. By combining RDPG embedding, Neural ODEs, and symbolic regression, we bridge the gap between black-box prediction and mechanistic understanding.

Theoretical foundations. The gauge-theoretic analysis (Section 2.2) provides principled answers to fundamental questions: *What can we learn?* All dynamics except uniform rotation around the origin (Theorem 4). *What architecture respects the structure?* The form $\dot{X} = N(P)X$ with symmetric N (Theorem 9). These results inform both architecture design and interpretation of learned models.

The gauge alignment challenge. We must be candid about a central difficulty: ASE introduces arbitrary gauge transformations at each time step—random $R^{(t)} \in O(d)$ from SVD sign ambiguity—that are unrelated to the dynamics. Even if true dynamics are perfectly smooth, the embedding trajectory $\widehat{X}^{(t)}$ jumps erratically. Existing joint embedding methods like UASE (Section 2.1.5) assume generative models incompatible with ODE dynamics on latent positions—they model time-varying "activity" against fixed "identity," not evolving positions.

Structure-constrained alignment as a path forward. Our proposed solution (Section 2.1.10) uses dynamics structure to identify and remove the random ASE gauge artifacts. The key insight: random gauge errors introduce skew-symmetric "contamination" that cannot be fit by symmetric dynamics (Theorem 3). By jointly optimizing over gauges and dynamics with symmetry constraints, we implicitly find $Q_t \approx \left(R^{(t)}\right)^{-1}$—the corrections that undo ASE artifacts. For linear horizontal dynamics ($\dot{X} = NX$ with N symmetric), we derive an alternating algorithm with closed-form steps and identifiability guarantees under generic conditions.

This is philosophically similar to how physical constraints help identify coordinates: we don't try to solve the alignment problem in isolation, but let the dynamics model guide the choice of gauge.

Practical obstructions. Beyond the theoretical gauge freedom, practical challenges include: (i) estimation error in $\widehat{X}$ from SVD ($\approx$35% position error, though $\hat{P}$ has only $\approx$5% error); (ii) finite-sample gauge jumps that make $\dot{\widehat{X}}$ undefined; (iii) discrete, noisy observations rather than continuous $P(t)$. The structure-constrained approach addresses (ii) by construction, but (i) and (iii) remain fundamental limitations of spectral methods.

Evaluation in $\boldsymbol{P}$-space. The gauge freedom implies that X-based metrics (e.g., position RMSE) are coordinate-dependent and potentially misleading. The natural evaluation metric is $P(t) = X(t)X(t)^\top$, which is gauge-invariant. Comparing predicted $\hat{P}(t)$ to true $P(t)$ directly tests whether the learned dynamics capture observable network structure, independent of coordinate conventions. Our experiments confirm that models can achieve low P-error even when X-trajectories differ substantially due to gauge ambiguity.

Parsimonious architectures. The $\dot{X} = N(P)X$ architecture with symmetric N offers two key advantages over generic neural networks: (1) automatic gauge consistency—symmetric N cannot produce invisible dynamics (Theorem 9), and (2) dramatic parameter reduction—polynomial N achieves comparable accuracy with 10^3–10^4 fewer parameters. When the true dynamics have this form, the polynomial parameterization can recover exact coefficients, providing interpretability that symbolic regression cannot match for black-box neural networks.

Model diagnostics. The tangent space constraint (Proposition 12) provides a diagnostic for model adequacy. If residuals $\dot{\hat{P}} - \dot{P}_{\text{pred}}$ have systematic structure in the null-null block $V_\perp^\top(\cdot)V_\perp$, this suggests either that the RDPG model is inappropriate or that the latent dimension d should be increased. This connects temporal network modeling to static dimension selection methods, potentially enabling dynamic diagnostics for detecting emerging community structure.

Limitations. The long-tailed network shows higher reconstruction error, suggesting challenges with highly heterogeneous degree distributions. The polynomial $N(P)X$ form, while parsimonious, may be too restrictive for dynamics that do not factor through P. For such cases, the kernel or general symmetric architectures provide a middle ground. Additionally, all methods rely on accurate RDPG embedding, which introduces estimation error ($\approx$35% in positions, though only $\approx$ 5% in probabilities).

Open problems. Several important questions remain:

1. **Sample complexity:** How many time points T and nodes n are needed for reliable structure-constrained alignment?
2. **Model selection:** How to choose the dynamics family $\mathcal{F}$ when the true form is unknown? Cross-validation on held-out time points may help.
3. **Beyond linear:** Can the alternating optimization approach extend efficiently to nonlinear dynamics families?
4. **Theoretical guarantees:** Under what conditions does the alternating algorithm converge to the global optimum?

Extensions. The UDE framework (Section 2.4) enables incorporating domain knowledge. For ecological networks, one might encode known trophic interactions in N_{known} while learning corrections. For social networks, community structure could inform block-diagonal parameterizations. The theory extends to directed graphs (Section 10), where $P = LR^\top$ with separate dynamics for source and target embeddings, and to oscillatory dynamics (Section 9), which symmetric N can produce through nonlinear coupling despite having real eigenvalues in the linear case.

6 Conclusion

We introduced a framework for learning interpretable dynamics of temporal networks, grounded in gauge-theoretic analysis of Random Dot Product Graphs.

What we established: The RDPG embedding has inherent $O(d)$ rotational ambiguity (gauge freedom). We characterized which dynamics are observable versus invisible under this symmetry: uniform rotations leave the probability matrix $P = XX^\top$ unchanged and cannot be detected. The architecture $\dot{X} = N(P)X$ with symmetric N is gauge-consistent by construction—it cannot produce invisible dynamics—and enables dramatic parameter reduction compared to generic neural networks.

What remains challenging: Aligning spectral embeddings across time to learn continuous dynamics is a hard open problem. Existing joint embedding methods (UASE, Omnibus) assume generative models incompatible with ODE dynamics on latent positions. We proposed structure-constrained alignment—joint optimization over gauges and dynamics using structural assumptions as regularization—and derived algorithms with identifiability guarantees for restricted dynamics families. This transforms the underdetermined problem into a tractable one, but practical effectiveness depends on having appropriate inductive bias and sufficient data.

The path forward: Learning continuous dynamics from discrete network snapshots requires either (a) working directly in gauge-invariant spaces (e.g., on P rather than X), or (b) using dynamics structure to regularize gauge choice. We developed the mathematical foundations for approach (b) and showed it can work for linear horizontal dynamics. Extension to more expressive dynamics families, characterization of sample complexity, and empirical validation on real temporal networks remain important directions.

Our approach produces interpretable differential equations governing network evolution, enabling both prediction and mechanistic insight—provided the gauge alignment challenge is adequately addressed for the dynamics class of interest.

7 Acknowledgments

8 Data and Code Availability

The `RDPGDynamics.jl` package and all data are available at [repository URL]. Experiments can be reproduced with: `julia --project scripts/reproduce_paper.jl`

Bibliography

[1] P. Holme and J. Saramäki, “Temporal networks,” *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012, doi: https://doi.org/10.1016/j.physrep.2012.03.001.

[2] M. Lucas, A. Morris, A. Townsend-Teague, L. Tichit, B. Habermann, and A. Barrat, “Inferring cell cycle phases from a partially temporal network of protein interactions,” *Cell Reports Methods*, vol. 3, no. 2, 2023.

[3] S. Hanneke, W. Fu, and E. P. Xing, “Discrete temporal models of social networks,” *Electronic Journal of Statistics*, vol. 4, no. none, pp. 585–605, 2010, doi: 10.1214/09-EJS548.

[4] T. Poisot, D. B. Stouffer, and D. Gravel, “Beyond species: why ecological interaction networks vary through space and time,” *Oikos*, vol. 124, no. 3, pp. 243–251, 2015.

[5] M. A. Porter and J. P. Gleeson, “Dynamical systems on networks,” *Frontiers in Applied Dynamical Systems: Reviews and Tutorials*, vol. 4. Springer, 2016.

[6] P. Holme, “Modern temporal network theory: a colloquium,” *The European Physical Journal B*, vol. 88, no. 9, p. 234, 2015.

[7] S. M. Kazemi *et al.*, “Representation learning for dynamic graphs: A survey,” *Journal of Machine Learning Research*, vol. 21, no. 70, pp. 1–73, 2020.

[8] A. Athreya *et al.*, “Statistical inference on random dot product graphs: a survey,” *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 8393–8484, 2018.

[9] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural ordinary differential equations,” in *Advances in neural information processing systems*, 2018.

[10] S. J. Young and E. R. Scheinerman, “Random dot product graph models for social networks,” in *Algorithms and Models for the Web-Graph: 5th International Workshop, WAW 2007*, 2007, pp. 138–149.

[11] I. Gallagher, A. Jones, and P. Rubin-Delanchy, “Spectral embedding for dynamic networks with stability guarantees,” in *Advances in Neural Information Processing Systems*, 2021, pp. 10158–10170.

[12] K. Levin, A. Athreya, M. Tang, V. Lyzinski, Y. Park, and C. E. Priebe, “A central limit theorem for an omnibus embedding of multiple random graphs and implications for multiscale network inference,” *arXiv preprint arXiv:1705.09355*, 2017.

[13] J. Arroyo, A. Athreya, J. Cape, G. Chen, C. E. Priebe, and J. T. Vogelstein, “Inference for multiple heterogeneous networks with a common invariant subspace,” *Journal of Machine Learning Research*, vol. 22, no. 142, pp. 1–49, 2021.

[14] P. Rubin-Delanchy, J. Cape, M. Tang, and C. E. Priebe, “Statistical inference on random dot product graphs: a survey,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 84, no. 4, pp. 1446–1476, 2022.

[15] C. Rackauckas *et al.*, “Universal Differential Equations for Scientific Machine Learning,” *CoRR*, 2020, [Online]. Available: https://arxiv.org/abs/2001.04385

[16] M. Schmidt and H. Lipson, “Distilling free-form natural laws from experimental data,” *science*, vol. 324, no. 5923, pp. 81–85, 2009.

9 Oscillatory Dynamics with Symmetric N

A common concern: can symmetric $N(P)$ produce oscillations? For a *linear* system $\dot{X} = NX$ with constant symmetric N, eigenvalues are real, so solutions are sums of exponentials—no oscillations.

However, our system $\dot{X} = N(P)X = N(XX^\top)X$ is *nonlinear* because N depends on X through P.

9.1 Linearization Around Equilibrium

At equilibrium X^* with $N(P^*)X^* = 0$, the linearization is:

$$\delta\dot{X} = N(P^*)\delta X + \left[\frac{\partial N}{\partial P}\,|_{P^*} \cdot \left(\delta X \cdot X^{*^\top} + X^* \cdot \delta X^\top\right)\right] X^* \tag{93}$$

The Jacobian (as a linear operator on $\delta X \in \mathbb{R}^{n\times d}$) is *not* simply $N(P^*)$. The second term involves derivatives of N and creates coupling that can produce complex eigenvalues.

9.2 Mechanisms for Oscillation

1. Hopf bifurcation: As parameters vary, eigenvalues of the Jacobian can cross the imaginary axis, creating limit cycles.

2. Amplitude-phase coupling (for $d = 2$): Write $X_i = r_i(\cos\theta_i, \sin\theta_i)$. Then $P_{ij} = r_i r_j \cos(\theta_i - \theta_j)$. Phase differences affect probabilities, which affect phase dynamics—a feedback loop enabling oscillation.

3. Multi-scale interaction:

$$N(P) = \alpha_1 P - \alpha_2 P^2 \tag{94}$$

Local attraction ($\alpha_1 P$) competes with nonlocal repulsion ($-\alpha_2 P^2$), potentially creating oscillatory approach to equilibrium.

9.3 What Symmetric N Cannot Do

- Rotation around origin in latent space (this is gauge/invisible anyway)
- Oscillations in the linear approximation with *constant* N

9.4 What Symmetric N Can Do

- Damped oscillations approaching equilibrium (via nonlinear Jacobian)
- Limit cycles via Hopf bifurcation
- Quasi-periodic motion in higher dimensions

10 Extension to Directed Graphs

For directed graphs, the probability matrix factors as $P = GR^\top$ where $G \in \mathbb{R}^{n\times d}$ contains source (giving) positions and $R \in \mathbb{R}^{n\times d}$ contains target (receiving) positions, with rows $G_{i\cdot} = \vec{g}_i^\top$ and $R_{i\cdot} = \vec{r}_i^\top$.

10.1 Gauge Group

The gauge transformation is $(G, R) \mapsto (GQ, RQ)$ for $Q \in O(d)$:

$$(GQ)(RQ)^\top = GQQ^\top R^\top = GR^\top = P \tag{95}$$

Crucially, both embeddings rotate by the *same* Q.

10.2 Gauge-Invariant Quantities

Under $(G, R) \mapsto (GQ, RQ)$:

- $P = GR^\top$ — invariant
- $\Gamma_G = GG^\top$ — invariant (source Gramian)
- $\Gamma_R = RR^\top$ — invariant (target Gramian)
- $G^\top G$, $R^\top R$, $G^\top R$ — transform by conjugation

The gauge-invariant data is (P, Γ_G, Γ_R)—three $n \times n$ matrices.

10.3 Equivariant Dynamics

Any $O(d)$-equivariant vector field has the form:

$$\dot{G} = N_{G(P,\Gamma_G,\Gamma_R)} \cdot G, \qquad \dot{R} = N_{R(P,\Gamma_G,\Gamma_R)} \cdot R \tag{96}$$

To eliminate gauge, require $N_G = N_G^\top$ and $N_R = N_R^\top$ individually.

10.4 Induced Dynamics on P

$$\dot{P} = N_G P + P N_R^\top \tag{97}$$

This is a Sylvester equation, enabling:

- Asymmetric growth patterns
- Directed community formation
- Source-target differentiation

Special case: If $N_G = N_R = N$ with N symmetric, the directed dynamics reduce to the undirected form $\dot{P} = NP + PN$.

11 Parsimonious UDE Parameterizations

This appendix catalogs parameterization choices for $N(P)$ in the UDE framework $\dot{X} = N(P)X$.

11.1 Taxonomy by Homogeneity

Type	**Description**	**Parameters**
Homogeneous	All node pairs follow same rule	$O(1)$ functions
Type-based	Nodes grouped into K types	$O(K^2)$ functions
Node-specific	Each node has own parameters	$O(n)$ scalars
Fully heterogeneous	Each pair has own parameter	$O(n^2)$—avoid!

Table 8: Parameterization complexity by homogeneity assumption.

11.2 Homogeneous Parameterizations

Polynomial in P:

$$N(P) = \alpha_0 I + \alpha_1 P + \alpha_2 P^2 + \ldots + \alpha_k P^k \tag{98}$$

Parameters: $k + 1$ scalars. Interpretation: $\alpha_0 I$ is self-dynamics, $\alpha_1 P$ is direct neighbor influence, $\alpha_2 P^2$ is two-hop influence.

Pairwise kernel:

$$N_{ij}(P) = \kappa(P_{ij}) \quad \text{for} \;\; i \neq j, \qquad N_{ii}(P) = h(P_{ii}) \tag{99}$$

Symmetry is automatic since $P_{ij} = P_{ji}$. The function κ can be a small neural network or parametric (e.g., $\kappa(p) = a + bp + cp^2$).

Attraction-repulsion (Lennard-Jones inspired):

$$\kappa(p) = \frac{a}{p + \varepsilon} - \frac{b}{(p + \varepsilon)^2} \tag{100}$$

Parameters: (a, b, ε). Equilibrium occurs where attraction balances repulsion.

Laplacian-based:

$$N(P) = \alpha\left(D^{-1/2} P D^{-1/2} - I\right) \tag{101}$$

where $D = \operatorname{diag}(P\mathbf{1})$ is the degree matrix. This encodes normalized diffusion.

11.3 Type-Based Parameterizations

When nodes belong to types $\tau : \{1, ..., n\} \to \{1, ..., K\}$, interactions can depend on type pairs.

Block kernel:

$$N_{ij} = \kappa_{\tau(i),\tau(j)}(P_{ij}) \tag{102}$$

Symmetry requires $\kappa_{ab} = \kappa_{ba}$. Parameters: $\frac{K(K+1)}{2}$ functions.

Stochastic Block Model prior:

$$N_{ij} = \alpha_{\tau(i),\tau(j)} + \beta \cdot P_{ij} \tag{103}$$

Parameters: $\frac{K(K+1)}{2}$ scalars α_{ab} plus one shared β.

Interpretation: Base rate depends on community pair, plus universal connection-strength effect.

11.4 Node-Specific Parameterizations

Each node has individual parameters, but interactions follow shared rules.

Diagonal + shared off-diagonal:

$$N_{ij} = \begin{cases} h_i & i = j \\ \kappa(P_{ij}) i \neq j \end{cases} \tag{104}$$

Parameters: n scalars h_i (node-specific self-rates) plus 1 function κ (shared interaction).

Node features determine rate:

$$h_i = g(\varphi_i), \quad \varphi_i = \left(P_{ii}, \sum_j P_{ij}, \max_j P_{ij}, ...\right) \tag{105}$$

where φ_i are node-level features extracted from P and g is a shared function (can be a small NN).

11.5 Message-Passing Formulation

An equivalent view writes dynamics as node-level updates:

$$\dot{X}_i = a(P_{ii})X_i + \sum_{j \neq i} m\big(P_{ij}\big)\big(X_j - X_i\big) \tag{106}$$

where a is the intrinsic rate and m is the message function.

The equivalent symmetric N is:

$$N_{ij} = \begin{cases} a(P_{ii}) - \sum_{k \neq i} m(P_{ik}) i = j \\ m\big(P_{ij}\big) & i \neq j \end{cases} \tag{107}$$

11.6 Low-Rank Parameterizations

Rank-r symmetric:

$$N = \sum_{k=1}^{r} \alpha_k u_k u_k^\top = U \ \mathrm{diag}(\alpha) U^\top \tag{108}$$

Parameters: $nr + r$ (with orthogonality constraints on U).

Data-derived basis:

$$N = \sum_{k=1}^{r} \alpha_k v_k v_k^\top \tag{109}$$

where v_k are the top eigenvectors of P itself. Parameters: r scalars only.

11.7 Encoding Qualitative Priors

Prior	**Parameterization**
Stability (nodes don't explode)	$N = -\exp(M)$ where $M = M^\top$
Conservation (tr(P) constant)	Project N to $\mathrm{tr}(NP) = 0$
Known equilibrium P^*	$N(P) = (P - P^*)M(P)$
Sparsity preservation	$N_{ij} = P_{ij} \cdot \kappa\big(P_{ij}\big)$

Table 9: Parameterizations encoding specific priors.

11.8 Geometric Boundary Constraints

There are two independent sources of constraints on $\dot{P}$: algebraic (rank preservation, automatically satisfied by $\dot{X} = N(P)X$) and geometric (probability bounds).

At lower boundary ($P_{ij} = 0$): Require $\dot{P}_{ij} \geq 0$.

From $\dot{P} = NP + PN$:

$$\dot{P}_{ij} = \sum_k N_{ik}P_{kj} + \sum_k P_{ik}N_{kj} \tag{110}$$

Caution: This is NOT simply a Metzler condition on N. For the linear system $\dot{y} = Ay$, Metzler A (non-negative off-diagonal) preserves the positive orthant. But $\dot{P} = NP + PN$ is a Lyapunov equation—the condition involves the entire structure of P, not just local properties of N.

At upper boundary ($\boldsymbol{P_{ij} = 1}$): Require $\dot{P}_{ij} \leq 0$.

Since $P_{ij} = X_i \cdot X_j \leq \|X_i\| \, \|X_j\|$, we have $P_{ij} = 1$ only if $X_i = X_j$ with $\|X_i\| = 1$.

Practical enforcement: Rather than modifying N (which breaks the symmetric structure) or projecting onto B_+^d (which distorts the geometry), use a barrier in the loss function:

$$\mathcal{L}_{\text{barrier}} = \gamma \sum_{i,j} \Big[\max\big(0, -P_{ij}\big)^2 + \max\big(0, P_{ij} - 1\big)^2 \Big] \tag{111}$$

This encourages learned dynamics to stay in the valid region while preserving the natural geometry of the embedding space. Our experiments show that projection-based constraint enforcement (e.g., onto B_+^d) is unnecessary and can actually impede learning.

Summary: In the interior of the valid configuration space, only the algebraic constraint matters—and it's automatic. Geometric constraints only matter at boundaries. Soft penalties (barrier loss) outperform hard constraints (projection) in practice.