Learning Continuous-Time Trajectories of Adolescent Brain Structural Connectomes via Latent Graph Neural ODEs: A Feasibility and Methodology Report

Executive Summary

The adolescent period represents a critical window of neurodevelopment characterized by non-linear structural reorganization, most notably the concurrent processes of synaptic pruning and myelination. These biological phenomena manifest macroscopically as changes in the structural connectome—the complex network of white matter tracts connecting gray matter regions. The Adolescent Brain Cognitive Development (ABCD) Study, with its large-scale longitudinal acquisition of multi-modal neuroimaging data, offers an unprecedented opportunity to map these trajectories. However, standard analytical frameworks in computational neuroscience often rely on discrete-time snapshots (e.g., annual visits), forcing continuous biological processes into rigid temporal bins and failing to account for the irregular sampling inherent in large clinical cohorts. This report presents a comprehensive technical feasibility analysis for a novel research proposal: "Learning Continuous-Time Trajectories of Adolescent Brain Structural Connectomes via Latent Graph Neural ODEs."

Our objective is to evaluate the viability of using **Latent Graph Neural Ordinary Differential Equations (LG-NODEs)** coupled with **Variational Graph Autoencoders (VGAEs)** to model the continuous evolution of brain networks. This approach shifts the paradigm from predicting static graph snapshots (G_{t+1} given G_t) to learning the underlying differential equations (dG/dt) governing connectome dynamics. This enables infinite temporal resolution, natural handling of missing data, and the potential to uncover the mechanistic "rules" of brain development.

The analysis synthesizes literature from geometric deep learning, Riemannian geometry, and topological data analysis. We identify that while standard Graph ODEs (e.g., *BrainODE*) successfully model signal propagation on *static* graphs, they are insufficient for modeling the *structural plasticity* required here. Consequently, we propose a **Coupled Graph Neural ODE (CG-ODE)** architecture where both node features (morphology) and edge weights (connectivity) evolve as coupled state variables. Furthermore, because structural connectomes reside on the Riemannian manifold of Symmetric Positive Definite (SPD) matrices, we recommend integrating **Diffeomorphic Conditional Flow Matching**

(DiffeoCFM) to ensure generated trajectories respect geometric constraints, superior to standard Euclidean operations.

We also address the critical need for topological validity. Traditional loss functions (MSE) often yield "blurry" graphs that lack the specific modularity and small-worldness of biological brains. We propose incorporating **Differentiable Persistent Homology** into the loss landscape to enforce topological consistency. Finally, we analyze the ABCD dataset's specific challenges—multi-modal fusion of sMRI and dMRI, and irregular sampling—and demonstrate how **Neural Controlled Differential Equations (Neural CDEs)** provide a robust mathematical solution for conditioning trajectories on sporadic observations.

This report concludes that the proposed research is not only feasible but represents a necessary methodological evolution. However, success depends on specific architectural choices: utilizing the adjoint sensitivity method for memory efficiency, enforcing manifold constraints via diffeomorphisms, and balancing reconstruction accuracy with topological fidelity.

---

1. Introduction: The Geometric and Temporal Complexity of Brain Development

The human brain is arguably the most complex network in existence, a dynamic system that undergoes continuous structural and functional reconfiguration throughout the lifespan. Adolescence, the period roughly spanning ages 10 to 19, is a particularly volatile and transformative epoch. It is characterized not merely by "growth" in the additive sense, but by a sophisticated refinement process involving the pruning of unused synaptic connections and the strengthening of active pathways through myelination. These micro-scale cellular changes aggregate to produce macro-scale alterations in the brain's structural connectome—the wiring diagram of the brain.

The Adolescent Brain Cognitive Development (ABCD) Study [1] has emerged as the definitive dataset for characterizing these changes. Enrolling over 11,800 children at ages 9-10 and following them for a decade, the study captures the trajectory of brain development with granularity previously impossible. The dataset includes multi-modal imaging: structural MRI (sMRI) providing morphological features like cortical thickness; diffusion MRI (dMRI) yielding structural connectivity (SC) matrices; and functional MRI (fMRI) capturing dynamic activity.

However, the richness of this data exposes the frailty of our current modeling tools. Neurodevelopment is a continuous-time process; a child's brain does not jump from a "Year 1 state" to a "Year 2 state" in a discrete hop. Yet, longitudinal analyses typically model it as such, treating scans as discrete snapshots. This introduces **discretization error** and makes the handling of **irregular sampling**—a ubiquitous issue where subjects return for follow-up at varying intervals (e.g., 14 months vs. 12 months)—mathematically awkward.

1.1 The Limitations of Discrete-Time Modeling

Current state-of-the-art approaches for longitudinal brain prediction often employ Recurrent Neural Networks (RNNs) or discrete-time Graph Neural Networks (GNNs). In these frameworks, the model learns a transition function F such that $G_{t+1} = F(G_t)$.

While effective for simple forecasting, this discrete formulation has fundamental limitations in the context of biological development:

1. **Temporal Rigidity:** They assume fixed time steps. If data is missing at $t=2$, it must be imputed or the subject discarded, discarding valuable information.
2. **Lack of Physical Insight:** They approximate the *state transition* rather than the *dynamics*. They tell us *where* the brain goes, but not the *rate* or *mechanism* of change.
3. **Topology Agnosticism:** Standard regression losses treat the adjacency matrix as a flat vector, ignoring the crucial graph topology—the cycles, hubs, and modules that define brain function.
4. **Geometric Violation:** Structural connectomes are positive definite matrices. Standard neural networks output Euclidean vectors, often producing mathematically invalid graphs that require post-hoc projection.[2]

1.2 The Promise of Continuous-Time Geometric Deep Learning

The proposal to use **Latent Graph Neural ODEs** fundamentally addresses these deficits. By parameterizing the derivative of the state $\frac{dZ}{dt} = f_\theta(Z, t)$, rather than the next state, Neural ODEs define a continuous trajectory that exists for all t. This allows for querying the brain state at any age (e.g., 12.5 years), naturally handling irregular observation times.

Moreover, by embedding this dynamics within a **Geometric Deep Learning** framework, we can respect the non-Euclidean nature of brain data. The structural connectome is not just a matrix; it is a point on a Riemannian manifold. The graph structure is not just an adjacency list; it is a topological object with specific homology. The proposed research aims to unify these perspectives—Dynamics (ODEs), Geometry (Manifolds), and Topology (Persistence)—into a single generative framework.

---

2. Comprehensive Literature Review and Theoretical Foundations

To assess the feasibility of the proposed "Latent Graph Neural ODE" framework, we must rigorously examine the existing literature across three distinct but converging fields:

Continuous-Time Graph Neural Networks, Generative Models for Graphs, and Computational Connectomics.

2.1 Neural Ordinary Differential Equations (Neural ODEs)

The introduction of Neural ODEs marked a paradigm shift in deep learning, replacing the discrete layers of a residual network with a continuous integration process.

2.1.1 The Fundamental Theorem of Neural ODEs

A standard Residual Network (ResNet) updates a hidden state h via:

$$h_{t+1} = h_t + f(h_t, \theta_t)$$

In the limit as the step size approaches zero, this difference equation becomes an ordinary differential equation:

$$\frac{dh(t)}{dt} = f(h(t), t, \theta)$$

Here, f is a neural network parameterized by θ. The output at any time T is obtained by a numerical ODE solver (e.g., Runge-Kutta):

$$h(T) = h(0) + \int_{0}^{T} f(h(t), t, \theta) dt$$

Relevance to Proposal: This formulation is naturally suited for longitudinal data. If a subject i has scans at t_1, t_2, t_3, the loss is simply the difference between the solver's state at those times and the observed data. The solver fills in the gaps continuously.[3]

2.1.2 The Adjoint Sensitivity Method

A critical feasibility concern for processing large brain graphs is memory. Backpropagating through an ODE solver naively requires storing all intermediate states of the integrator. The Adjoint Sensitivity Method 4 allows for calculating gradients by solving a second, augmented ODE backwards in time.

$$\frac{d \lambda(t)}{dt} = -\lambda(t)^T \frac{\partial f}{\partial h}$$

This decouples memory usage from the number of integration steps, making it feasible to model complex, long-term developmental trajectories of high-dimensional connectomes (e.g., 300+ ROIs) on standard GPU hardware.

2.2 Graph Neural ODEs (GDEs)

Extending ODEs to graph-structured data creates Graph Neural ODEs (GDEs). The literature presents two distinct classes of GDEs, and distinguishing between them is vital for this proposal.

2.2.1 Continuous Diffusion on Static Graphs (GRAND)

Models like GRAND (Graph Neural Diffusion) 5 interpret the message-passing mechanism of GNNs as a discretization of the heat diffusion equation:

$$\frac{\partial H(t)}{\partial t} = \text{div}(A(H) \nabla H(t))$$

These models are excellent for node classification and smoothing signals on a fixed graph structure.

- **Critique:** These are insufficient for the user's proposal. Adolescent development involves the *changing* of the graph structure itself (edge weights change). A model that assumes a static adjacency matrix A cannot capture synaptic pruning.

2.2.2 Latent Graph ODEs (LG-ODE)

Latent Graph ODEs [6] encode the graph into a latent vector Z, evolve Z via an ODE, and decode it back.

- **Mechanism:**
  1. Encoder: $z_0 = \text{GNN}(G_{t_0})$
  2. Dynamics: $\frac{dz}{dt} = f(z, t)$
  3. Decoder: $\hat{G}_t = \text{Dec}(z_t)$
- **Feasibility:** This approach is more promising. It abstracts the discrete graph changes into a continuous latent trajectory. However, it risks losing the explicit relational structure during the evolution phase if f is just a standard MLP.

2.2.3 Coupled Graph ODEs (CG-ODE)

The most relevant prior work is **Coupled Graph ODEs (CG-ODE)** [8], used for modeling multi-agent interacting systems.

- Core Innovation: It explicitly models the co-evolution of node features H and the adjacency matrix A.

  $$\frac{dH}{dt} = f_{node}(H, A, t)$$
  $$\frac{dA}{dt} = f_{edge}(H, A, t)$$
- **Application to Brains:** This mirrors biological reality.
  - $H(t)$: Morphological changes (cortical thinning).
  - $A(t)$: White matter connectivity changes.
  - The equations allow changes in cortical thickness to drive changes in connectivity, and vice versa. This "coupling" is the theoretical key to a successful model of the

ABCD data.

2.3 Neural Controlled Differential Equations (Neural CDEs)

A major challenge in the ABCD dataset is **irregular sampling**. Standard ODEs define an initial value problem (IVP): given z_0, predict the future. However, if we observe data at $t=0, t=2, t=5$, we want to update our trajectory based on the observations at $t=2$ and $t=5$, not just extrapolate from $t=0$.

Neural CDEs 5 solve this by treating the observed data X as a control path.

$$dZ(t) = f(Z(t)) dX(t)$$

The data stream $X(t)$ is interpolated (e.g., via cubic splines) to be continuous. The latent state Z evolves in response to changes in X.

- **Superiority:** Literature suggests Neural CDEs significantly outperform Neural ODEs on irregular time series because they can "read" the incoming data continuously rather than just predicting it.[11] For the ABCD study, where a subject's trajectory might be influenced by external factors measured at varying times (e.g., onset of substance use), CDEs offer a more rigorous formulation.

2.4 Generative Models: From VGAE to Flow Matching

To generate the brain graph at time t, we need a decoder that maps the latent state z_t to a graph G_t.

2.4.1 Variational Graph Autoencoders (VGAE)

VGAEs [12] are the standard for static graph generation. They learn a latent distribution $q(z|G)$ and sample from it.

- **Limitation:** The standard inner-product decoder ($A = \sigma(ZZ^T)$) assumes a specific type of community structure (homophily) and often fails to capture complex topologies.
- **Dynamic Extension:** Combining VGAE with ODEs (Motion-based VAE) allows the latent prior to be time-dependent $p(z_t | z_{t-1})$.

2.4.2 Riemannian Flow Matching (DiffeoCFM)

A critical recent development is **Flow Matching** on manifolds.[2]

- **The Manifold Problem:** Structural connectivity matrices are Symmetric Positive Definite (SPD). Interpolating between two SPD matrices linearly (Euclidean average) results in a matrix with "swelling" determinant—it adds energy (connections) that shouldn't exist.
- **DiffeoCFM Solution:** This method uses a diffeomorphism (bijective differentiable map)

ϕ to map the SPD manifold $\mathcal{M}$ to a Euclidean space $\mathcal{E}$ (Tangent space).
    - $\phi(A) = \log(A)$ (Matrix Logarithm).
    - Flow Matching is performed in $\mathcal{E}$ to learn a vector field $u_t(x)$.
    - The result is mapped back: $A = \exp(x)$.
- **Generative Quality:** DiffeoCFM is currently State-of-the-Art (SOTA) for generating brain connectomes, outperforming GANs and standard Diffusion models in maintaining geometric validity.[2]

2.5 Topological Data Analysis (TDA)

Traditional losses (MSE) struggle to capture global graph properties. Two graphs can have low MSE but radically different global topologies (e.g., one is fully connected, one is a ring).

- **Persistent Homology:** TDA tracks the birth and death of topological features (connected components H_0, loops H_1, voids H_2) as a filtration threshold is swept across the graph.
- **Differentiable TDA:** Recent work [15] has made the calculation of persistence diagrams differentiable. This allows us to construct a loss function $\mathcal{L}_{topo}$ that penalizes the generator if the Betti numbers of the generated graph do not match the ground truth. This is crucial for maintaining the "small-world" nature of brain networks during generation.

---

3. Methodology: The "Latent Graph Neural ODE" Framework

Based on the theoretical analysis, we articulate the methodology for the proposed research. This framework is designed to ingest the irregular, multi-modal ABCD data and produce continuous trajectories of structural connectomes.

3.1 Problem Formalization

Let $\mathcal{D} = \{ (t_j^{(i)}, G_{j}^{(i)}, M_{j}^{(i)}, C^{(i)}) \}$ be the dataset for N subjects.

- $i \in \{1, \dots, N\}$: Subject index.
- $t_j^{(i)}$: Age of subject i at visit j (continuous value, e.g., 9.4 years).
- $G_{j}^{(i)}$: Structural Connectome (Adjacency matrix $A \in \mathbb{R}^{R \times R}$) at visit j.
- $M_{j}^{(i)}$: Node Morphology Features matrix (Cortical Thickness, Area) at visit j.
- $C^{(i)}$: Static Covariates (Sex, Genetics, PDE status).

Objective: Learn a generative model $p_\theta(G(t) | G(t_0), M(t_0), C)$ that predicts the continuous trajectory of the connectome $G(t)$ for any t.

3.2 Proposed Architecture: The Manifold-Coupled Latent CDE

We propose a hybrid architecture that integrates **VGAE**, **Neural CDEs**, and **Riemannian Geometry**. The model consists of three stages: Diffeomorphic Encoding, Latent Coupled Dynamics, and Riemannian Decoding.

3.2.1 Stage 1: The Diffeomorphic Encoder

We cannot feed raw SPD matrices into a standard neural network without violating geometry.

1. Manifold Projection: We first map the SPD connectome A to the Euclidean tangent space via the matrix logarithm:

   $$L = \log(A)$$

   L is a symmetric matrix but lives in a vector space where Euclidean operations are valid.
2. Graph Encoding: We use a specialized GNN (e.g., Graph Transformer) to encode the geometry and morphology:

   $$Z_{node} = \text{GraphTransformer}(L, M)$$
   $$Z_{graph} = \text{Readout}(Z_{node})$$

   This produces a latent distribution $q(z_{t_0} | G_{t_0}, M_{t_0}) = \mathcal{N}(\mu, \sigma)$.

3.2.2 Stage 2: Latent Coupled Dynamics (The Core)

This component models the *continuous evolution* of the brain. We employ a **Coupled Neural CDE** to handle the interaction between local morphology and global connectivity.

We define two coupled latent states: $z_{node}(t)$ (representing regional features) and $z_{adj}(t)$ (representing connectivity structure).
The evolution is governed by:

$$\frac{d z_{node}(t)}{dt} = f_{\theta_1}(z_{node}, z_{adj}, t, C)$$

$$\frac{d z_{adj}(t)}{dt} = f_{\theta_2}(z_{node}, z_{adj}, t, C)$$

Handling Irregular Sampling via CDE:
Instead of a pure autonomous ODE, we use the CDE formulation where the derivative depends on the interpolated path of the observed data $X(t)$ (constructed from the sparse visits via cubic splines):

$$dZ(t) = f_\theta(Z(t)) dX(t)$$

This allows the model to update its trajectory whenever new data is observed (e.g., at Year 2 visit), correcting the prediction path. This is mathematically superior to RNNs which would require "padding" missing years with zeros.

3.2.3 Stage 3: The Riemannian Decoder

The solver outputs $z_{adj}(t)$ at the query time t. We must reconstruct the connectome.

1. **Euclidean Reconstruction:** An MLP maps $z_{adj}(t)$ back to the tangent space matrix $\hat{L}_t$.
2. Manifold Projection (Inverse): We apply the exponential map to return to the SPD manifold:

   $$\hat{A}_t = \exp(\hat{L}_t)$$

   This step guarantees that the generated connectome is always positive definite, a critical constraint often ignored in simpler models.

3.3 Loss Function Landscape

Training generative models on graphs is notoriously unstable. Minimizing edge-wise MSE leads to blurry "average" brains. We propose a composite loss function:

$$\mathcal{L} = \mathcal{L}_{Recon} + \lambda_{KL} \mathcal{L}_{KL} + \lambda_{Topo} \mathcal{L}_{Topo} + \lambda_{Smooth} \mathcal{L}_{Smooth}$$

1. Riemannian Reconstruction Loss ($\mathcal{L}_{Recon}$): instead of Frobenius norm, we use the Log-Euclidean distance, which is the geodesic distance on the manifold:
   $$ \mathcal{L}_{Recon} = |

| \log(A_{GT}) - \log(\hat{A}{Pred}) ||F^2 $$
2. **Topological Consistency Loss ($\mathcal{L}{Topo}$):** This addresses the requirement for "Topological Consistency." We compute the Persistence Diagrams (PD) of the ground truth and predicted graphs. The loss is the Wasserstein Distance between these diagrams:
$$\mathcal{L}{Topo} = W_p(\text{PD}(A_{GT}), \text{PD}(\hat{A}{Pred}))$$
This forces the generator to match the number of cycles and connected components, preserving the "fingerprint" of the brain's topology.
3. Sparsity & Smoothness:
* $\mathcal{L}{Smooth}$: Kinetic energy regularization on the ODE path to prevent chaotic trajectories.
* $\mathcal{L}_{Sparsity}$: L_1 regularization on the adjacency matrix to encourage biologically realistic sparsity (brains are not fully connected).

---

4. Domain Application: Longitudinal Brain Network Prediction

This section contrasts the proposed methodology with alternative approaches within the specific context of the ABCD study domain.

4.1 Graph Generation vs. Regression

- **Regression (Baseline):** A model that outputs A_{t+1} directly.
  - *Flaw:* Deterministic. It predicts the "mean" future brain. It cannot capture the *distribution* of possible developmental trajectories.
- **Generative Modeling (Proposed):** By using a VGAE-based latent space, we model $p(G_{t+1}|G_t)$.
  - *Advantage:* We can sample multiple possible futures for a single child. This is crucial for identifying "deviant" trajectories (e.g., early signs of psychopathology) by comparing the child's actual development against the distribution of generated healthy possibilities.

4.2 Diffusion/Flow Matching vs. ODEs

This is a critical architectural decision.

- **Score-Based Generative Models (Diffusion/Flow Matching):** These are currently the gold standard for *image* generation quality. They transform noise into data via a stochastic differential equation (SDE) or probability flow ODE.
  - *Pros:* Extremely high fidelity samples. DiffeoCFM [14] handles the SPD constraint perfectly.
  - *Cons:* They are computationally expensive to sample (requires solving an ODE for *every* generation step). More importantly, they typically model the distribution of *static* data, not temporal *trajectories*.
- **Latent Graph ODEs (Proposed):** These model the *time evolution* in latent space.
  - *Verdict:* For *longitudinal prediction* (what will this specific child look like in 2 years?), Latent ODEs are superior because they explicitly model the temporal dynamics $f(z,t)$. Flow matching models the *density* $p(x)$.
  - *Hybrid Proposal:* We recommend using **Latent ODEs for the trajectory** and **Flow Matching for the Decoder**. The Latent ODE predicts the future latent code z_t, and a conditional Flow Matching decoder generates the high-fidelity graph from z_t. This combines the dynamic strengths of ODEs with the generative strengths of Flow Matching.

---

5. Data & Features: The ABCD Study Context

The Adolescent Brain Cognitive Development (ABCD) Study is the largest long-term study of brain development in the US. Implementing this model on ABCD data requires handling specific idiosyncrasies.

5.1 Dataset Characteristics

- **Subjects:** ~11,800 children, recruited at ages 9-10.
- **Longitudinal Design:** Annual visits. Currently, data up to Year 4 or 5 is available.
- **Atlas:** We assume the use of the **Desikan-Killiany** atlas (84 ROIs) or **Destrieux** atlas (148 ROIs). Using voxel-level connectivity is computationally infeasible for ODEs.
- **Modalities:**
  - **dMRI:** Used to construct Structural Connectivity (SC). Preprocessing involves deterministic or probabilistic tractography (e.g., using MRTrix3) to generate streamline counts.
  - **sMRI:** T1-weighted images processed via FreeSurfer to yield ROI-wise cortical thickness, surface area, and volume.

5.2 Handling Irregular Sampling

This is the "killer application" for ODEs/CDEs.

- **The Problem:** In clinical reality, visits are not perfectly spaced. Child A visits at months 0, 12, 24. Child B visits at months 0, 14, 23. Standard RNNs require binning data into discrete slots (e.g., "Year 1"), introducing error.
- **The ODE Solution:** The ODE solver operates in continuous time. We feed the exact age t_{age} into the solver.
  - Input: $\{(t=9.2, G_0), (t=10.4, G_1)\}$
  - The loss is computed exactly at $t=9.2$ and $t=10.4$. The solver integrates over the interval $[9.2, 10.4]$. No imputation or binning is required.

5.3 Multi-Modal Fusion

The integration of sMRI and dMRI is critical. Structural connectivity (dMRI) does not change in isolation; it is driven by cortical maturation (sMRI).

- **Fusion Strategy:** We draw inspiration from **NeuroKoop** [17], which fuses sMRI and fMRI using a Koopman operator.
- **Proposed Mechanism:**
  1. Construct a **Heterogeneous Graph** or simply augment node features.
  2. Nodes V have features $x_v =$.
  3. Edges E have features $e_{uv} =$.
  4. The **Coupled ODE** updates node features based on neighbors (sMRI evolution driven by connectivity) and updates edge weights based on endpoint nodes (connectivity

driven by cortical thinning). This bidirectional coupling captures the biological reality of Hebbian plasticity and pruning.

---

6. Metrics & Topology Constraints

Evaluating generative models of brain graphs requires more than pixel-wise accuracy.

6.1 Sparsity

Brain networks are sparse (density $\approx 5-15\%$).

- **Constraint:** Generative models (like VGAE) often produce fully connected, weighted graphs with many near-zero edges.
- **Solution:** Implement **Top-K filtering** or learn a dynamic threshold parameter within the decoder. The L_1 regularization term in the loss function ($\lambda ||A||_1$) also encourages sparsity.

6.2 Topological Consistency (Persistent Homology)

This is the most advanced metric. We must ensure the "shape" of the predicted brain is correct.

- **Metric:** We calculate the **Betti Numbers** (β_0: connected components, β_1: cycles) of the generated graphs.
- **Loss Implementation:** We use the **Wasserstein Distance** between the persistence diagrams of the generated graph $\hat{G}$ and the ground truth G.
  - Libraries like torchph (PyTorch Persistent Homology) [18] allow for backpropagating through this calculation.
  - This explicitly penalizes the model if it breaks a major white matter tract (creating a new connected component) or fuses distinct modules (destroying a cycle).

6.3 Temporal Smoothness

Development is continuous.

- **Metric:** The discrete second derivative of the trajectory.
- **Constraint:** We add a regularization term $\int ||\frac{d^2z}{dt^2}|| dt$ to the ODE loss. This penalizes erratic, jagged trajectories, enforcing the biological prior that brain development is a smooth process (though the *rate* may change).

---

7. Technical Recommendations and Potential Pitfalls

Based on the synthesis of the user's requirements and the literature, we provide specific technical recommendations.

7.1 Architecture Roadmap

Component	Recommendation	Justification
Input Representation	**Log-Euclidean Space**	Map SPD matrices A to symmetric matrices $L = \log(A)$ to allow unconstrained neural processing.
Latent Dynamics	**Neural CDE**	Superior to ODEs for irregular time series; allows the trajectory to be conditioned on the observed data path.
Coupling	**Node-Edge Co-evolution**	Use the formulation: $dH/dt = \text{GNN}(H, A)$, $dA/dt = \text{MLP}(H_{src}, H_{dst})$ to capture structure-function interplay.
Decoder	**Flow Matching Head**	Use a conditional Flow Matching model to decode latent state z_t to high-fidelity manifold data, rather than a simple MLP.
Loss	**Topo-Geometric Loss**	Combine Log-Euclidean distance (Geometry) with Wasserstein Persistence Loss (Topology).
Solver	**Dopri5 (Runge-Kutta 4/5)**	Adaptive step size is essential. Brain development has "stiff" dynamics (rapid changes

		vs plateaus).

7.2 Computational Pitfalls and Mitigations

Pitfall	Description	Mitigation Strategy
Computational Complexity of Topology	Calculating Persistent Homology is $O(N^3)$ or worse. Doing this every training step is prohibitive.	**Stochastic Topology:** Compute topological loss only on a random subset of batches or every k epochs. Use approximate topological measures.
Memory Explosion	Backprop through ODE solver stores the entire trajectory graph.	**Adjoint Method:** Use torchdiffeq.odeint_adjoint. This solves the ODE backwards to reconstruct gradients, using constant $O(1)$ memory wrt time steps.
Manifold Validity	Generated matrices might drift off the SPD manifold if not constrained.	**Diffeomorphisms:** Strictly enforce the $\exp(\cdot)$ map at the output. Never train directly on raw adjacency coefficients without this map.
Vanishing Gradients in Long Sequences	Integrating over 10 years of development might lead to gradient decay.	**Neural CDE:** The control path $X(t)$ re-injects gradient signal at every observation point, mitigating the vanishing gradient problem inherent in pure ODEs.

7.3 Data Pitfalls

- **Confounding Variables:** Age effects in the ABCD study are confounded by scanner upgrades and site differences.

- *Mitigation:* Include "Site ID" and "Scanner Version" as static covariates in the ODE function $f(z, t, \text{covariates})$. This allows the model to "explain away" site effects as a bias term in the derivative.

---

8. Conclusion

The proposed research, "Learning Continuous-Time Trajectories of Adolescent Brain Structural Connectomes via Latent Graph Neural ODEs," represents a sophisticated integration of cutting-edge Geometric Deep Learning with high-impact Computational Neuroscience. By moving from discrete regression to continuous dynamic modeling, it directly addresses the fundamental limitations of current longitudinal analyses: irregular sampling, discretization error, and topological blindness.

The feasibility analysis confirms that the theoretical building blocks—Graph ODEs, Neural CDEs, and Diffeomorphic Flow Matching—are mature enough to be integrated. The primary challenge lies in the engineering of the **Coupled** dynamics (ensuring nodes and edges drive each other) and the computational cost of the **Topological Loss**. However, with the adoption of the Adjoint method and efficient TDA libraries, these obstacles are surmountable.

This project has the potential to produce the first "generative simulator" of adolescent brain development, capable of interpolating missing scans, forecasting future trajectories, and identifying personalized deviations indicative of neuropsychiatric risk with unprecedented geometric and topological fidelity.

---

9. Key Literature Table

Category	Paper / Model	Key Contribution & Relevance
Graph ODEs	**LG-ODE** [6]	Introduced the concept of evolving graphs in a latent space using ODEs. Foundational.
Graph ODEs	**CG-ODE** [8]	**Coupled Graph ODE.** Explicitly models the co-evolution of nodes and

		edges (dA/dt). Critical for structural plasticity modeling.
Irregular Series	**Neural CDE** [5]	**Graph Neural CDE.** Solves the irregular sampling problem by treating data as a continuous control path.
Brain Dynamics	**BrainODE** [19]	Applied ODEs to fMRI BOLD signals. Demonstrates feasibility on brain data, though limited to static graphs.
Generative	**DiffeoCFM** [2]	**Riemannian Flow Matching.** SOTA for generating SPD matrices (connectomes). Recommended for the decoder component.
Multi-Modal	**NeuroKoop** [17]	Demonstrates fusion of sMRI and fMRI in ABCD data using operator theory.
Topology	**Diff. Topology** [15]	**Differentiable Persistent Homology.** Enables training with topological loss functions (Wasserstein distance on persistence diagrams).
Topology	**TopoSinGAN** [21]	Shows practical implementation of topological loss in a generative adversarial framework.

(Note: This report assumes the use of Python 3.9+, PyTorch 2.0+, PyTorch Geometric,

torchdiffeq, and torchph for implementation.)

Works cited

1. Research Publications - ABCD Study, accessed December 10, 2025, https://abcdstudy.org/publications/page/15/
2. Riemannian Flow Matching for Brain Connectivity Matrices via Pullback Geometry, accessed December 10, 2025, https://www.researchgate.net/publication/392105400_Riemannian_Flow_Matching_for_Brain_Connectivity_Matrices_via_Pullback_Geometry
3. BrainODE: Neural Shape Dynamics for Age- and Disease-aware Brain Trajectories - OpenReview, accessed December 10, 2025, https://openreview.net/pdf/880ef14ea12d26d9ce56b714466baf6faf211f69.pdf
4. Graph Neural Stochastic Differential Equations, accessed December 10, 2025, https://www.mlmi.eng.cam.ac.uk/files/2022_-_2023_dissertations/graph_neural_stochastic_differential_equations.pdf
5. Learning Dynamic Graph Embeddings with Neural Controlled Differential Equations, accessed December 10, 2025, https://www.researchgate.net/publication/368716803_Learning_Dynamic_Graph_Embeddings_with_Neural_Controlled_Differential_Equations
6. Coupled Graph ODE for Learning Interacting System Dynamics - NSF PAR, accessed December 10, 2025, https://par.nsf.gov/servlets/purl/10260000
7. (PDF) A Social Dynamical System for Twitter Analysis - ResearchGate, accessed December 10, 2025, https://www.researchgate.net/publication/390177216_A_Social_Dynamical_System_for_Twitter_Analysis
8. PI-NeuGODE: Physics-Informed Graph Neural Ordinary Differential Equations for Spatiotemporal Trajectory Prediction - IFAAMAS, accessed December 10, 2025, https://www.ifaamas.org/Proceedings/aamas2024/pdfs/p1418.pdf
9. Coupled Graph ODE for Learning Interacting System Dynamics - UCLA Computer Science, accessed December 10, 2025, https://web.cs.ucla.edu/~yzsun/papers/2021_KDD_CG_ODE.pdf
10. Learning Dynamic Graph Embeddings with Neural Controlled Differential Equations - arXiv, accessed December 10, 2025, https://arxiv.org/html/2302.11354v2
11. Permutation Equivariant Neural Controlled Differential Equations for Dynamic Graph Representation Learning - arXiv, accessed December 10, 2025, https://arxiv.org/html/2506.20324v2
12. Dynamic Node Privacy Feature Decoupling Graph Autoencoder Based on Attention Mechanism - MDPI, accessed December 10, 2025, https://www.mdpi.com/2076-3417/15/12/6489
13. AutoGraphAD: A novel approach using Variational Graph Autoencoders for anomalous network flow detection - arXiv, accessed December 10, 2025, https://arxiv.org/html/2511.17113v1
14. Riemannian Flow Matching for Brain Connectivity Matrices via Pullback Geometry

- arXiv, accessed December 10, 2025, https://arxiv.org/html/2505.18193v2
15. Neural Approximation of Graph Topological Features, accessed December 10, 2025, https://proceedings.neurips.cc/paper_files/paper/2022/file/d7ce06e9293c3d8e6cb3f80b4157f875-Paper-Conference.pdf
16. Topology-aware Graph Diffusion Model with Persistent Homology - OpenReview, accessed December 10, 2025, https://openreview.net/forum?id=ZC0wgCabT2
17. NeuroKoop: Neural Koopman Fusion of Structural–Functional Connectomes for Identifying Prenatal Drug Exposure in Adolescents - arXiv, accessed December 10, 2025, https://arxiv.org/html/2508.16414v1
18. c-hofer/torchph: The essence of my research, distilled for reusability. Enjoy - GitHub, accessed December 10, 2025, https://github.com/c-hofer/torchph
19. BrainODE: Dynamic Brain Signal Analysis via Graph-Aided Neural Ordinary Differential Equations | AI Research Paper Details - AIModels.fyi, accessed December 10, 2025, https://www.aimodels.fyi/papers/arxiv/brainode-dynamic-brain-signal-analysis-via-graph
20. BrainODE: Dynamic Brain Signal Analysis via Graph-Aided Neural Ordinary Differential Equations - Emory CS, accessed December 10, 2025, https://www.cs.emory.edu/~jyang71/files/brainode.pdf
21. TopoSinGAN: Learning a Topology-Aware Generative Model from a Single Image - MDPI, accessed December 10, 2025, https://www.mdpi.com/2076-3417/14/21/9944?type=check_update&version=3