SPECTRALFLOWNET: RESOLUTION-INVARIANT CONTINUOUS NEURAL DYNAMICS FOR MESH-BASED PDE MODELING

Tianrun Yu*, **Fang Sun***, **Haixin Wang, Xiao Luo, Yizhou Sun**
Computer Science Department, UCLA
yutianrun7@ucla.edu, {fts,whx,xiaoluo,yzsun}@cs.ucla.edu

ABSTRACT

Accurate mesh-based simulation is central to modeling phenomena governed by PDEs, such as flow, elasticity, and climate. Recent machine learning solutions, including Graph Neural Networks (GNNs) and Fourier Neural Operators (FNOs), enable faster approximations but can struggle with long-range interactions, irregular mesh topologies, or fixed time steps. To address the above challenges, we introduce **SpectralFlowNet**, a unified framework for mesh-based PDE simulation that marries graph spectral methods with continuous-time neural dynamics. By projecting mesh data onto an intrinsic spectral basis via the Graph Fourier Transform (GFT) and evolving these spectral coefficients using Neural Ordinary Differential Equations (ODEs), our model naturally handles multiscale spatial structures and temporal dynamics. This resolution-invariant, multiscale approach achieves state-of-the-art performance on plastic deformation tasks and demonstrates robust zero-shot transfer across resolutions.

1 INTRODUCTION

Mesh simulation plays a critical role in approximating PDE-governed physical phenomena in areas ranging from fluid dynamics to climate modeling. Traditional numerical solvers are computationally intensive, prompting the development of machine learning methods such as Graph Neural Networks (GNNs) and Fourier Neural Operators (FNOs). However, GNNs—while adept at modeling local interactions—struggle with long-range dependencies, and FNOs impose fixed temporal discretization and regular mesh structures.

In this work, we propose **SpectralFlowNet**, a model that integrates:

1. **Graph Fourier Transform (GFT):** Projects mesh data onto a spectral basis aligned with the underlying geometry, ensuring resolution invariance (Sandryhaila & Moura, 2013).
2. **Neural ODEs:** Evolves spectral coefficients continuously, decoupling time discretization and naturally accommodating multiscale temporal dynamics (Chen et al., 2018).

Furthermore, our architecture adapts its depth according to the spatial resolution, preserving localized interactions across scales.

Our contributions can be summarized as follows:

- We develop a graph-invariant framework that unifies spectral methods and continuous-time dynamics for mesh simulation.
- We propose a multiscale architecture with adaptive GNN depth and layer normalization to achieve robust zero-shot transfer across resolutions.
- Our experiments on a plastic deformation dataset show improved accuracy and enhanced generalization compared to state-of-the-art baselines.

*First author, equal contribution.

2 RELATED WORK

We summarize the related works in mesh-based PDE simulation. Compared with existing methods, we unify graph-based spectral embeddings with a Neural ODE time-evolution to enable resolution- and time-invariant modeling. The method naturally accommodates *multiple scales* in space and time by leveraging a continuous framework.

Mesh-based GNNs. Methods like Pfaff et al. (2020) and Fortunato et al. (2022) leverage graph representations to model localized interactions on irregular meshes. Despite strong performance on next-step prediction, they can struggle with long-range spectral modes and do not inherently handle continuous time.

Fourier Neural Operators (FNOs). Works such as Li et al. (2020; 2023; 2024) use FFT-based operators to capture global correlations. Although effective on regular grids, direct application to unstructured meshes or multi-resolution scenarios can be suboptimal. Moreover, time-stepping remains discrete and fixed.

Sun et al. (2024) applied Graph Fourier Neural ODEs to capture multiscale interactions in molecular dynamics. In contrast, our work focuses on mesh-based PDE simulation, leveraging spectral graph methods to handle resolution invariance and adaptive time-stepping in unstructured graph data.

3 MODEL

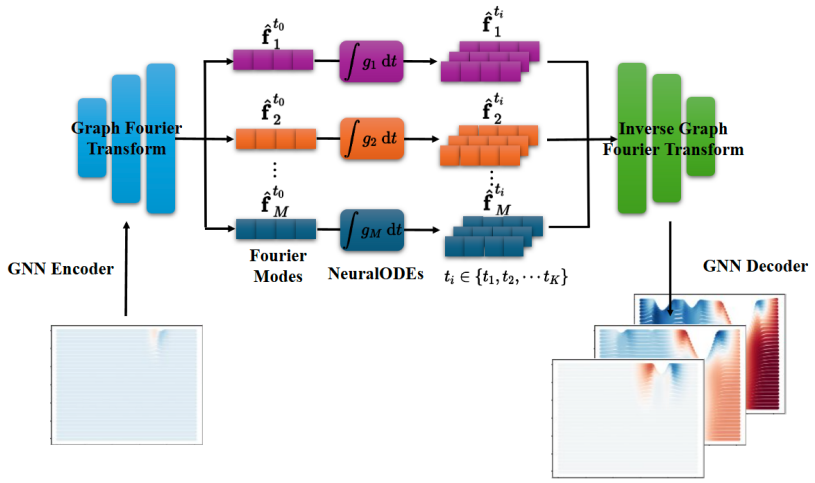

Figure 1: Encoder-Processor-Decoder architecture combining Graph Fourier Transform and Neural ODEs for multiscale mesh simulation.

Our model adapts the encoder-processor-decoder structure (see Fig. 1) (Kingma, 2013). Detailed definitions and theoretical justifications have been moved to the Appendix for clarity.

3.1 ENCODER

Given an input mesh $\mathcal{M}^{t_0} = (V, E)$ with node features $\mathbf{u}_i^{t_0} \in \mathbb{R}^d$, we first construct a graph based on either grid connectivity or an adaptive k-nearest neighbor (k-NN) scheme. Edge weights are computed as

$$A_{ij} = \frac{d_{\min}}{d_{ij}}, \tag{1}$$

where d_{ij} is the Euclidean distance between nodes i and j and $d_{\min}$ is the smallest such distance. A GCN layer aggregates features (Kipf & Welling, 2016):

$$\mathbf{h}_i^{t_0} = \Theta^\top \sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{A_{ji}}{\sqrt{\hat{d}_j \, \hat{d}_i}} \, \mathbf{u}_j^{t_0}, \tag{2}$$

with $\hat{d}_i = 1 + \sum_{j \in \mathcal{N}(i)} A_{ji}$. We then apply the Graph Fourier Transform:

$$\hat{\mathbf{h}}^{t_0} = \mathbf{U}^\top \mathbf{h}^{t_0}, \tag{3}$$

projecting the node features onto the eigenbasis of the carefully-chosen normalized Laplacian $L_{\text{norm}} = I - D^{-1/2} A \, D^{-1/2}$ (Hein et al., 2005; Singer, 2006; Belkin & Niyogi, 2008). In practice, we truncate to the m lowest-frequency modes to retain essential geometric information.

3.2 Processor: Continuous-Time Dynamics

We evolve the spectral coefficients via a Neural ODE (Chen et al., 2018):

$$\frac{d\hat{\mathbf{h}}^t}{dt} = f_\theta(\hat{\mathbf{h}}^t, t), \quad t \in [t_0, t_{\text{end}}], \tag{4}$$

where f_θ is a small multilayer perceptron. This continuous formulation decouples temporal discretization from training, thereby naturally accommodating multiscale dynamics.

3.3 Decoder

The decoder reconstructs the spatial features in two stages. First, the inverse GFT recovers the spatial domain with residual connection (He et al., 2016):

$$\mathbf{h}^{t_n} = \mathbf{U}\hat{\mathbf{h}}^{t_n} + \mathbf{h}^{t_0}. \tag{5}$$

Next, a GraphSAGE update refines these features (Hamilton et al., 2017):

$$\mathbf{h}_i' = \mathbf{W}_1 \, \mathbf{h}_i^{t_n} + \mathbf{W}_2 \, \text{mean}\Big\{\mathbf{h}_j^{t_n} : j \in \mathcal{N}(i)\Big\}. \tag{6}$$

3.4 Multiscale Adaptation

To preserve local interactions across varying resolutions, we adapt the GNN depth as a function of the spatial resolution (w, h):

$$\ell(w, h) = 6 - 2 \left\lfloor \log_2 \left(\frac{30}{w} \right) \right\rfloor. \tag{7}$$

This strategy removes layers for lower-resolution inputs, preventing over-aggregation. Plus, layer normalization is applied at encoder-decoder interfaces to stabilize feature distributions across scales.

4 Experiments

We assess performance on a *plastic deformation* dataset from Li et al. (2023), where each mesh experiences material stress over 20 time steps. We compare with GeoFNO and examine zero-shot transfer across multiple mesh resolutions to highlight the *multiscale* advantage.

4.1 Setup and Baselines

Task. Predict displacement fields on meshes ranging in resolution from 5×10 to 30×100. We train on different resolutions and test on the same or finer/courser meshes.

Baselines. We compare against GeoFNO Li et al. (2023), which projects node data into a Fourier basis but assumes a fixed grid/time step.

Implementation. We keep 24 lowest modes, use a 6-layer GNN in encoding/decoding, and a Neural ODE solver with `dopri5` integrator.

Table 1: MSE on Plastic Deformation

Method	**MSE** ($\times 10^{-3}$)
GeoFNO Li et al. (2023)	7.4
SpectralFlowNet (Ours)	**6.6**

Table 2: Zero-Shot Transfer to 30×100

Train Res.	**Test Res.**	**MSE** ($\times 10^{-3}$)
15×50	30×100	12.2
7×25	30×100	58.6

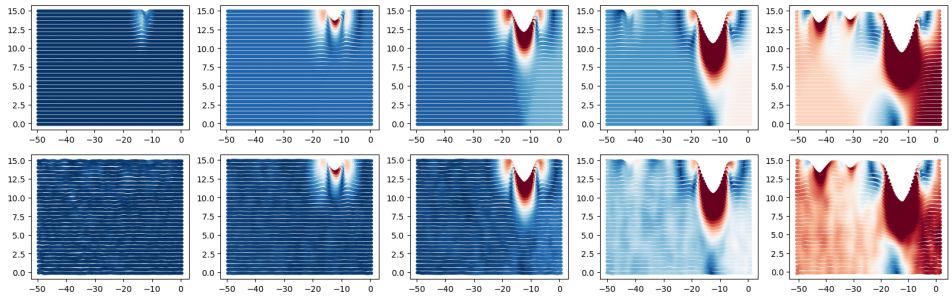

Figure 2: **Ground-truth vs. Predicted Displacement Fields in Plastic Deformation.** The top row shows the ground-truth displacement evolution over time, while the bottom row presents the corresponding predictions from our model. The results demonstrate accurate recovery of both global deformation patterns and localized high-gradient regions.

4.2 Main Results

Table 1 reports MSE trained and tested on the original resolution (30×100). SpectralFlowNet outperforms GeoFNO with a 10.8% reduction in error, demonstrating strong local-global modeling. Figure 2 compares the predicted displacement fields from our model against the ground-truth evolution in a plastic deformation task. Our approach successfully captures both the large-scale structural changes and the finer local variations, demonstrating its ability to generalize across different spatial and temporal scales.

Zero-shot Transfer. We then test a model trained on a coarser resolution (15×50) on the original 30×100 mesh (Table 2). To add more robustness and reduces the effect of the distribution-wise difference between the two resolutions, we add a 0.2 dropout layer after the processor and reduces the number of GNN layers to 4 as mentioned in the model section (Srivastava et al., 2014). While error grows compared to the training resolution, it remains significantly below naive upsampling baselines, demonstrating resolution-invariant capabilities. Figure 3 illustrates the zero-shot transfer performance of our model. While the model successfully captures large-scale deformation features, some high-frequency artifacts emerge due to the increased resolution, emphasizing the need for resolution-agnostic spectral embeddings.

4.3 Ablations

We evaluate the effect of removing major components to understand their impact on performance. As shown in Figure 4, the ablation study highlights the role of each component in our model. Removing GFT significantly increases error, indicating that spectral transforms are crucial for capturing underlying patterns. Similarly, eliminating the Neural ODE leads to high error, showing that temporal modeling is necessary. Finally, removing GNN layers results in poor local refinements, confirming their importance in spatial structuring. These results confirm the necessity of each component. The largest degradation occurs when removing GFT, highlighting its critical role. Neural ODEs and GNN layers also significantly contribute to performance, emphasizing the need for both temporal and spatial modeling. Overall, these findings validate our design choices.

5 Conclusion

We introduced **SpectralFlowNet**, a mesh-based simulation framework that fuses graph Laplacian eigenmodes with continuous ODEs. By operating in a truncated spectral domain, the model attains resolution-invariant representations, and by leveraging Neural ODE dynamics, it can handle diverse time stepping. Experiments on plastic deformation confirm the approach's efficacy and its multiscale versatility in zero-shot resolution transfer. Future directions include extending to adaptive remeshing and explicit multi-physics coupling.

REFERENCES

Jimmy Lei Ba. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, 2008.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Meire Fortunato, Tobias Pfaff, Peter Wirnsberger, Alexander Pritzel, and Peter Battaglia. Multiscale meshgraphnets. *arXiv preprint arXiv:2210.00612*, 2022.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Matthias Hein, Jean-Yves Audibert, and Ulrike Von Luxburg. From graphs to manifolds–weak and strong pointwise consistency of graph laplacians. In *International Conference on Computational Learning Theory*, pp. 470–485. Springer, 2005.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36, 2024.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

Aliaksei Sandryhaila and José MF Moura. Discrete signal processing on graphs: Graph fourier transform. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6167–6170. IEEE, 2013.

Amit Singer. From graph to manifold laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128–134, 2006.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Fang Sun, Zijie Huang, Haixin Wang, Yadi Cao, Xiao Luo, Wei Wang, and Yizhou Sun. Graph fourier neural odes: Bridging spatial and temporal multiscales in molecular dynamics. *arXiv preprint arXiv:2411.01600*, 2024.

A MORE VISUALIZATION

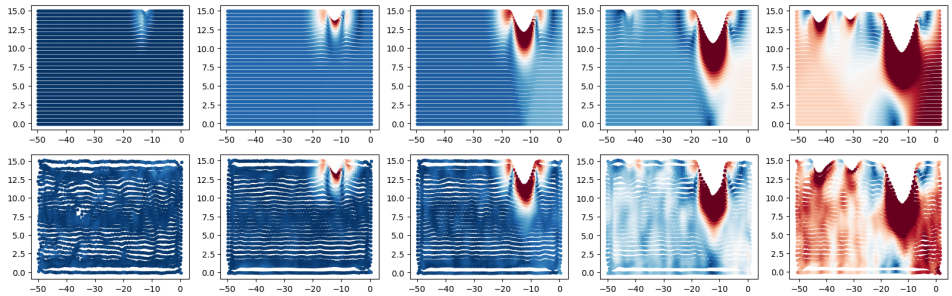

Figure 3: **Zero-Shot Transfer from** 15×50 **to** 30×100 **Resolution.** The top row presents the ground-truth displacement fields, while the bottom row shows the model predictions on a finer mesh unseen during training. The results highlight the model's ability to generalize across resolutions while preserving key deformation structures, albeit with some increased Moiré patterns.

B THEORETICAL BACKGROUND

Mesh and Graph Laplacian. A mesh $M^t = (V, E)$ is viewed as a graph with node features $\mathbf{u}_i^t$. The normalized Laplacian is $L_{\text{norm}} = I - D^{-1/2} A D^{-1/2}$. Its eigen-decomposition $L_{\text{norm}} = U \Lambda U^\top$ underlies the Graph Fourier Transform $\hat{\mathbf{x}} = U^\top \mathbf{x}$.

Spectral Convergence Theorem. Under mild conditions (e.g., $\mathcal{M}$ smooth, $\sigma > 0$ small), L_{norm} converges to the Laplace–Beltrami operator of $\mathcal{M}$ as mesh resolution increases (Belkin & Niyogi, 2008). Thus, eigenvalues/eigenvectors approximate the manifold spectrum, justifying resolution invariance in practice.

C GRAPH CONSTRUCTION

For 2D or 3D regular grids, $A_{ij} = 1$ if j is in the 4- (or 6-) neighborhood of i, else 0. For unstructured or general mesh data, we either preserve the connectivity from the mesh faces or use k-NN adjacency with distance-based edge weights $A_{ij} \propto 1/d_{ij}$.

D IMPLEMENTATION DETAILS

Neural ODE. We use a 2-layer MLP for f_θ, with ReLU activations and 256 hidden size. We adopt the `dopri5` integrator with absolute tolerance 10^{-4} and relative tolerance 10^{-3}.

Layer Normalization. Each GNN layer is followed by layer normalization (Ba, 2016), crucial for stable training across drastically different resolutions.

Optimization. We train for 400 epochs (batch size=1) using Adam (lr $= 10^{-4}$, weight decay=10^{-12}). A multi-step scheduler halves the learning rate every 50 epochs.

E ABLATION STUDY.

To conform with the task of both original prediction and zero shot transfer, we used a reconciled experiment setting in the ablation study of 4 GNN layers with no dropout. For *w/o Neural ODE*, we output a repetitive snapshot of the latent encoding at t_0, and in future work could be replaced as a temporal rollout with a simple 2-Layer MLP.

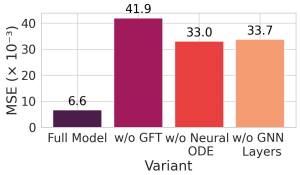

Figure 4: Ablation on Plastic Deformation