Topological Zigzag Spaghetti for Diffusion-based Generation and Prediction on Graphs

Yulia R. Gel

Virginia Tech

joint with

Yuzhou Chen
University of California-Riverside and Princeton University

Outline

Motivation

- Most currently available deep learning (DL) architectures are *inherently static* and do not systematically integrate time-dimension into the learning process.
- Such static approaches then require more frequent model updates and are limited in their ability to adapt to time-evolving environment.
- In contrast, many recent studies show that topological summaries may contain important complementary information on the underlying data generating process and deliver insights on the suitable DL architectures, which results in enhanced performance and robustness of DL models.

Why Topological data analysis (TDA)?

Three key advantages:

- Topology studies shapes in a *coordinate-free* manner
- Topology studies properties of shapes that are invariant to *minor perturbations*
- Topology deals with *compressed representations of shapes*

TDA is the emerging methodology at the interface of algebraic topology, computational mathematics, statistics, and data science:

- Systematically infers qualitative and quantitative patterns from the data directly.
- Enhances our understanding on the latent role of topology and geometry in organization and functionality of complex systems.
- One of its main tools, **persistent homology**, studies data shape through a multi-resolution view of nested *simplicial complexes* over the data cloud.

What if the Shape Changes over Time? And it does!

In the standard PH framework, we look at the object through multiple lenses. However, given a sequence of time-evolving graphs (or other objects) $\mathcal{G}^{t_1}, \mathcal{G}^{t_2}, \ldots, \mathcal{G}^{t_n}$, we may be interested in such questions as:

- **Q1** *What are the most characteristic topological signatures of these graph sequence over time?*
- **Q2** *How do these time-aware topological signatures vary over different resolution scales?*
- **Q3** *How certain are we that the extracted over time and over different scales α_k most characteristic topological signatures are indeed most characteristic and are not simply due to a chance only?*

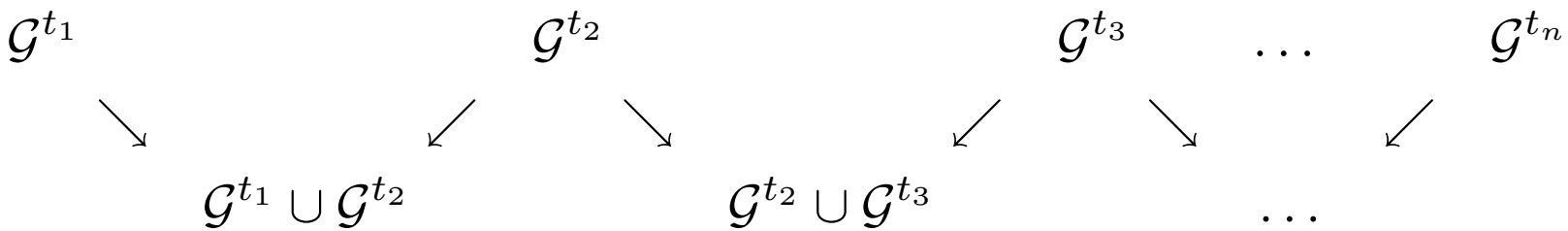
These questions are critical for forecasting, transfer learning, anomaly detection, and other unsupervised tasks on time-evolving objects.

Time-Aware Zigzag Persistence

We can invoke the machinery of zigzag persistence (ZP) (Carlsson & Silva, 2010) and consider linear maps into both directions

$$\mathscr{K}(\mathcal{G}_{\alpha_k}) \hookrightarrow \mathscr{K}(\mathcal{G}_{\alpha_{k+1}}) \text{ and } \mathscr{K}(\mathcal{G}_{\alpha_k}) \hookleftarrow \mathscr{K}(\mathcal{G}_{\alpha_{k+1}}).$$

For time-evolving graphs, we alternate left and right inclusions and interleave them with unions of neighboring graphs:



NB: ZP is much more general than PH for time-evolving objects, based on the diamond principle and rooted in quiver representations.

Time-Aware Zigzag Persistence (contd)

Now, given a scale α, we say that a topological feature ξ of dimension p $(0 \leq p \leq dim(\mathcal{K}))$

- is born at time t_b, if it is first recorded at $\mathscr{K}(\mathcal{G}^{t_b})$, and born at time $t_b + \frac{1}{2}$, if it is first recorded at $\mathscr{K}(\mathcal{G}^{t_b} \cup \mathcal{G}^{t_b+1})$.
- dies at t_d or $t_d + \frac{1}{2}$, if it is last recorded at $\mathscr{K}(\mathcal{G}^{t_d})$ or $\mathscr{K}(\mathcal{G}^{t_d} \cup \mathcal{G}^{t_d+1})$, respectively.

The extracted topological features at scale α_k can be represented in a form of a **zigzag persistent diagram (ZPD)**

$$\mathrm{PDz}_{\alpha_\mathrm{k}} = \{(\mathrm{b}_\xi, \mathrm{d}_\xi) \in \mathbb{R}^2 | \mathrm{b}_\xi < \mathrm{d}_\xi, \xi \in \mathcal{M}\}, \;\; \mathrm{k} = 1, 2, \ldots, \mathrm{m},$$

where b_ξ and d_ξ are the birth and death of ξ, respectively. Here $\mathcal{M}$ is a set containing the observed p-dimensional topological features at scale α_k, and m is a filtration length.

Time-Aware Zigzag Persistence (contd)

To quantify topological features extracted over time, we can

- Use zigzag persistence images (ZPI) (Chen et al., 2021) and zigzag filtration curves (ZFC) (Chen et al., 2022), based on advancing the ideas of persistence images (Adams et al, 2017) and filtration curves (Johnson & Jung, 2021, O'Bray et al, 2021), developed for a traditional PH to a zigzag case.

  However, ZPI and ZFC are limited **only** to a **single pre-defined** resolution scale α_*. And we do not know good α_*!

- Alternatively, we can use a crocker plot (Xian et al., 2020), which does not use the notion of ZP, but uses the traditional PH framework, recording the number of "holes" at each scale α_i as a function of time t and scales $\alpha_1, \alpha_2, \ldots$.

  Crocker plot is simple but is it is **not a differentiable summary** and, hence, cannot serve as an input to a fully trainable topological layer in DL models.

Can we address these challenges and get the best from both?

Time-Aware Zigzag Persistence: Zigzag Spaghetti (ICLR'25)

Definition (Zigzag Spaghetti)

Let $[t_1, t_n]$ be the time interval over which we observe $\{\mathcal{G}^t\}_{t_1}^{t_n}$. We represent $[t_1, t_n]$ as $\cup \Delta t_i$, where $\Delta t_i = (t_{i-1} + \frac{1}{2}, t_i), i = 1, \ldots, n$ are non-overlapping time intervals. Let α_k, $k = 1, \ldots, m$ be a sequence of scales. Then, Zigzag Spaghetti (ZS) is given by

$$ZS\big(\{\mathcal{G}^t\}_t\big) = \begin{bmatrix} \sum_{j=1}^{\mathcal{M}} \omega_1 \kappa_1^{\alpha_1}(t_{b_j}, t_{d_j})_{\alpha_1} & \cdots \sum_{j=1}^{\mathcal{M}} \omega_n \kappa_n^{\alpha_1}(t_{b_j}, t_{d_j})_{\alpha_1} \\ \vdots & \vdots \\ \sum_{j=1}^{\mathcal{M}} \omega_1 \kappa_1^{\alpha_m}(t_{b_j}, t_{d_j})_{\alpha_m} & \cdots \sum_{j=1}^{\mathcal{M}} \omega_n \kappa_n^{\alpha_m}(t_{b_j}.t_{d_j})_{\alpha_m} \end{bmatrix}$$

Here $\kappa_i^{\alpha_k} : \mathbb{R}^2 \mapsto \mathbb{R}$ is a suitable Lipschitz continuous function with Lipschitz constant L_i, associated α_k, $k = 1, 2, \ldots, m$.

As κ_i, we use a Gaussian density f with mean $(t_{i-1} + 1/2, t_i)$ and $\Sigma = \mathbb{I}$; $(t_{b_j}, t_{d_j})_{\alpha_k} \in \mathbb{R}^2$ is an interval containing the birth and death of topological feature ξ_j observed at α_k, $j = \{1, 2, \ldots, \mathcal{M}\}$ and $\omega_i > 0$ such that $\sum_i \omega_i = 1$. (In our studies we set $\omega_i = 1/n$, i.e. a "flat prior".)

Stability Properties of ZS (ICLR'25)

Theorem (Stability Guarantees of ZS)

Let PDz_{α_k} *be a zigzag persistence diagram corresponding to scale* α_k, $k = 1, 2, \ldots, m$, *and let* PDz'_{α_k} *be its perturbed counterpart such that* $\mathcal{W}_1\big(PDz_{\alpha_k}, PDz'_{\alpha_k}\big) < \epsilon_k$. *Let* ZS *and* ZS' *be ZS summaries corresponding to* PDz_{α_k} *and* PDz'_{α_k}, *respectively, over scales* $\alpha_1, \alpha_2, \ldots, \alpha_m$. *Then ZS is stable with respect to Wasserstein distance* $\mathcal{W}_1$ *and*

$$||ZS - ZS'||_\infty \leq \max_{1 \leq k \leq m} \mathcal{W}_1\big(PDz_{\alpha_k}, PDz'_{\alpha_k}\big),$$

where $||\cdot||_\infty$ *is a column norm of a matrix, i.e. for a matrix* B, $||\mathbf{B}||_\infty = \max_i \sum_j |b_{ij}|$.

Bootstrap over Zigzag Spaghetti (ICLR'25)

Employ ZP for topological uncertainty quantification (UQ). Given a sequence of graphs $\{\mathcal{G}^t\}_{t_1}^{t_n}$ and its associated ZS,

- (sub)sample τ_m graphs $\{\mathcal{G}^\tau\}_{\tau_1}^{\tau_n}$ without replacement out of t_n graphs $\{\mathcal{G}^t\}_{t_1}^{t_n}$ $(\tau_m < t_n)$
- construct its associated ZS^*
- repeat the (sub)sampling procedure B times, which results in an ensemble of bootstrapped ZS (BZS):

$$\mathrm{BZS} = \{\mathrm{ZS}_1^*, \mathrm{ZS}_2^*, \ldots, \mathrm{ZS}_{\mathrm{B}}^*.\}$$

Intuitively, we can expect that the most illustrative time-aware topological features persisting over $\{\mathcal{G}^t\}_{t_1}^{t_n}$ shall also manifest in many bootstrapped ZS.

Armed with the BZS ensemble, we can then consider integrating BZS into DL models, quantifying the uncertainty associated with time-aware topological signatures.

The Idea of Zigzag Spaghetti-Diffusion Models in a Nutshell (ICLR'25)

Step 1 The **forward diffusion process** is to gradually add noise onto the real data.

Here, to capture topological information both from graph $\mathcal{G}$ and its node features, we construct a mixed-up graph $\mathcal{G}_{\mathscr{M}} = (\boldsymbol{A}_{\mathscr{M}}, \boldsymbol{X})$ based on original input graph $\mathcal{G}_{\mathcal{O}} = (\boldsymbol{A}_{\mathcal{O}}, \boldsymbol{X})$ and $\mathcal{K}$-hop graph $\mathcal{G}_{\mathcal{K}} = (\boldsymbol{A}_{\mathcal{K}}, \boldsymbol{X})$.

First, to capture graph structural information of nodes in topology and feature spaces, we build a $\mathcal{K}$-hop graph via (i) cosine similarity; (ii) Gaussian kernel; and (iii) node embedding similarity. Then both the ordinary graph $\mathcal{G}_{\mathcal{O}}$ and $\mathcal{K}$-hop graph $\mathcal{G}_{\mathcal{K}}$ are fed into the graph representation learning module for obtaining node latent embeddings, i.e.,

$$\boldsymbol{Z}_{\mathcal{G}_{\mathcal{O}}}^{(\ell+1)} = f_{\mathrm{MLP}}\left(\sigma\left(\hat{\boldsymbol{A}}_{\mathcal{O}}^{\tau}\boldsymbol{H}_{\mathcal{G}_{\mathcal{O}}}^{(\ell)}\boldsymbol{\Theta}_{\mathcal{O}}^{(\ell)}\right)\right),$$
$$\boldsymbol{Z}_{\mathcal{G}_{\mathcal{K}}}^{(\ell+1)} = f_{\mathrm{MLP}}\left(\sigma\left(\hat{\boldsymbol{A}}_{\mathcal{K}}^{\tau}\boldsymbol{H}_{\mathcal{G}_{\mathcal{K}}}^{(\ell)}\boldsymbol{\Theta}_{\mathcal{K}}^{(\ell)}\right)\right),$$

The Idea of Zigzag Spaghetti-Diffusion Models in a Nutshell (ICLR'25)

Step 1 The **forward diffusion process** is to gradually add noise onto the real data.

Here, to capture topological information both from graph $\mathcal{G}$ and its node features, we construct a mixed-up graph $\mathcal{G}_{\mathscr{M}} = (\boldsymbol{A}_{\mathscr{M}}, \boldsymbol{X})$ based on original input graph $\mathcal{G}_{\mathcal{O}} = (\boldsymbol{A}_{\mathcal{O}}, \boldsymbol{X})$ and $\mathcal{K}$-hop graph $\mathcal{G}_{\mathcal{K}} = (\boldsymbol{A}_{\mathcal{K}}, \boldsymbol{X})$.

Given $\boldsymbol{Z}_{\mathcal{G}_{\mathcal{O}}}$ and $\boldsymbol{Z}_{\mathcal{G}_{\mathcal{K}}}$, we adopt node-level attention mechanism to learn the hidden connectivity between nodes and build the corresponding mixed-up graph which can be expressed as

$$\boldsymbol{e}_{uv}^{\mathscr{M}} = \boldsymbol{\Theta}_{\mathscr{M}}[\boldsymbol{Z}_{\mathcal{G}_{\mathcal{O}}}, \boldsymbol{Z}_{\mathcal{G}_{\mathcal{K}}}],$$

$$\alpha_{e_{uv}^{\mathscr{M}}} = \text{Softmax}(\boldsymbol{e}_{uv}^{\mathscr{M}}) = \frac{\exp(\sigma(\boldsymbol{P}_{\mathscr{M}} \boldsymbol{e}_{uv}^{\mathscr{M}}))}{\sum_{v' \in \mathcal{V}} \exp(\sigma(\boldsymbol{P}_{\mathscr{M}} \boldsymbol{e}_{uv'}^{\mathscr{M}}))},$$

where $\alpha_{e_{uv}^{\mathscr{M}}}$ denotes the weight of the edge between nodes u and v.

The Idea of Zigzag Spaghetti-Diffusion Models in a Nutshell (ICLR'25)

Step 1 The **forward diffusion process** is to gradually add noise onto the real data.

Here, to capture topological information both from graph $\mathcal{G}$ and its node features, we construct a mixed-up graph $\mathcal{G}_{\mathscr{M}} = (\boldsymbol{A}_{\mathscr{M}}, \boldsymbol{X})$ based on original input graph $\mathcal{G}_{\mathcal{O}} = (\boldsymbol{A}_{\mathcal{O}}, \boldsymbol{X})$ and $\mathcal{K}$-hop graph $\mathcal{G}_{\mathcal{K}} = (\boldsymbol{A}_{\mathcal{K}}, \boldsymbol{X})$.

Next, to generate a ZS for the sample $\boldsymbol{X}_t$ at time step t, we apply ZS over both time step t and its adjacent φ ($\varphi > 2$) time steps, e.g.,

$$\mathcal{T}_{t-\varphi:t+\varphi} = \{t-\varphi, \ldots, t-2, t-1, t, t+1, t+2, \ldots, t+\varphi\}.$$

For UQ, we also randomly select B subsamples from $\mathcal{T}_{t-\varphi:t+\varphi}$.

Finally, We generate a noised sample $\boldsymbol{X}_t$ from the input node feature $\boldsymbol{X}$ (i.e., $\boldsymbol{X}_0$) and learn ZS from the mixed-up graph (i.e., $\boldsymbol{Z}_{\mathsf{ZS},t}$):

$$\boldsymbol{X}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{X}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon,$$
$$\boldsymbol{Z}_{\mathsf{ZS},t} = f_{\mathsf{ZS}}(f_{\mathsf{MGC}}(\boldsymbol{X}_t)).$$

Idea of Zigzag Spaghetti-Diffusion Models in a Nutshell: cont'd (ICLR'25)

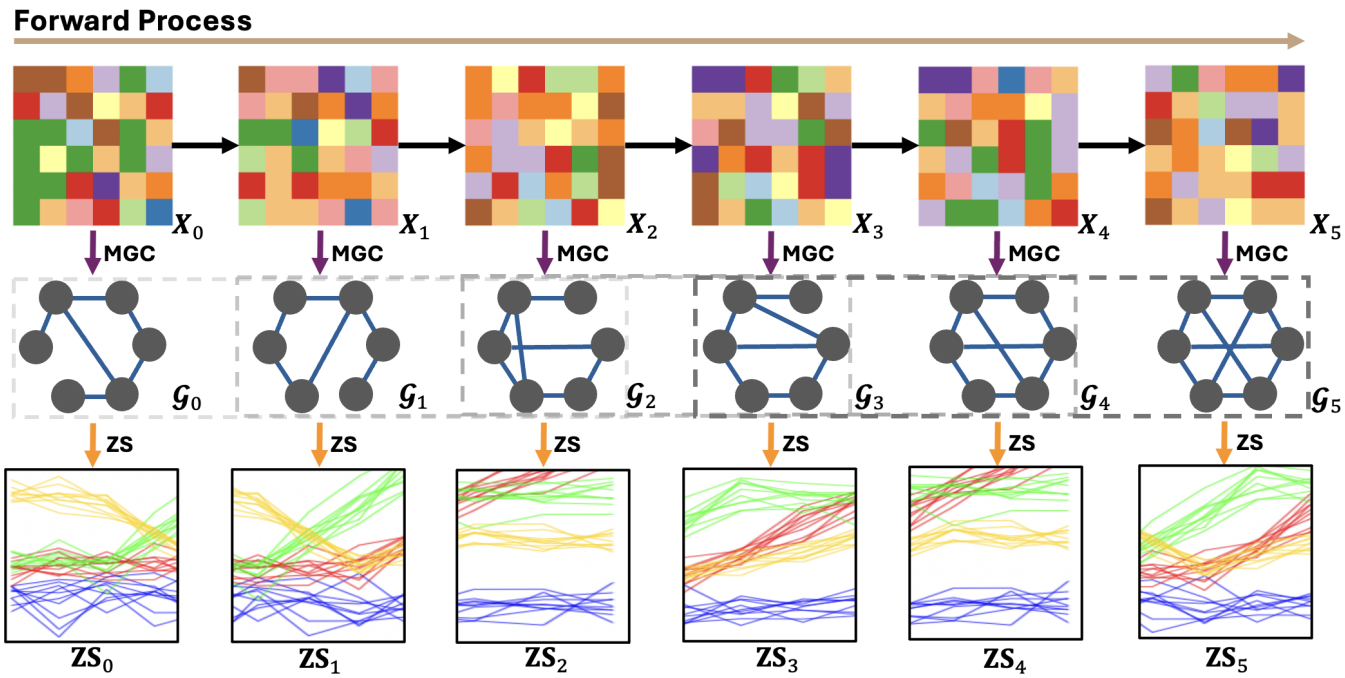
Step 2 The **reverse diffusion process** is to make the model learn to denoise the data.

We develop a denoising decoder $f_{\text{DEC}}(\cdot)$ with graph convolution blocks to learn the reverse Markov chain with zigzag spaghetti:

$$\tilde{\boldsymbol{X}}_0 = f_{\text{DEC}}(\boldsymbol{X}_t, \boldsymbol{Z}_{\text{ZS},t}, t) = [f_{\text{GNN}}(\boldsymbol{X}_t), f_{\text{ZS}}(\boldsymbol{Z}_{\text{ZS-ENC},t})] + f_{\text{PE}}(t),$$

where sinusoidal function $f_{\text{PE}}(\cdot)$ is used to encode timestep t (i.e., positional encoding).

Zigzag Spaghetti-Diffusion Model (ZS-DM) (ICLR'25)



Forward process of ZS-DM:

1. At each time step, we add noise to the data, i.e., transitioning from $\boldsymbol{X}_0$ to $\boldsymbol{X}_1$ to $\boldsymbol{X}_2$,..
2. Then, we use the mixed-up graph construction (MGC) approach to generate the corresponding new mixed-up graph.
3. Finally, we apply ZS over the mixed-up graphs within a specific sliding window (i.e., the dashed gray box) to obtain the corresponding ZS.

Experiments of ZS-DM (ICLR'25)

We validate our ZS-DM on unsupervised graph representation learning [1,2,3] and spatio-temporal forecasting tasks [4] on

1. **3 chemical compound datasets:** MUTAG, BZR, and COX2
2. **5 molecular compound datasets:** NCI1, D&D, PROTEINS, PTC_MR, and PTC_FM
3. **1 Open Graph BenchMark (OGB):** ogbg-molhiv
4. **2 traffic datasets:** PeMSD3 and PeMSD8

Experimental Results: Graph Classification (ICLR'25)

Performance comparison on molecular and chemical graphs

Model	NCI1	PROTEINS	DD	MUTAG	BZR	COX2	PTC_MR	PTC_FM
GL [12]	N/A	N/A	N/A	81.66±2.11	N/A	N/A	57.30±1.40	N/A
WL [13]	80.01±0.50	72.92±0.56	74.00±2.20	80.72±3.00	N/A	N/A	58.00±0.50	N/A
DGK [19]	80.31±0.46	73.30±0.82	N/A	87.44±2.72	N/A	N/A	60.10±2.60	N/A
node2vec [3]	54.89±1.61	57.49±3.57	N/A	72.63±10.20	N/A	N/A	N/A	N/A
sub2vec [1]	52.84±1.47	53.03±5.55	N/A	61.05±15.80	N/A	N/A	N/A	N/A
graph2vec [8]	73.22±1.81	73.30±2.05	N/A	83.15±9.25	N/A	N/A	N/A	N/A
InfoGraph [14]	76.20±1.06	74.44±0.31	72.85±1.78	89.01±1.13	84.84±0.86	80.55±0.51	61.70±1.40	61.55±0.92
GraphCL [21]	77.87±0.41	74.39±0.45	78.62±0.40	86.80±1.34	84.20±0.86	81.10±0.82	61.30±2.10	65.26±0.59
AD-GCL [15]	73.91±0.77	73.28±0.46	75.79±0.87	88.74±1.85	85.97±0.63	78.68±0.56	63.20±2.40	64.99±0.77
RGCL [6]	78.14±1.08	75.03±0.43	78.86±0.48	87.66±1.01	84.54±1.67	79.31±0.68	61.43±2.50	64.29±0.32
GCL-TAGS [7]	71.43±0.49	75.78±0.41	75.78±0.52	89.12±0.76	N/A	N/A	N/A	N/A
GraphMAE [5]	80.40±0.30	75.30±0.39	N/A	88.19±1.26	N/A	N/A	N/A	N/A
CWN [2]	80.16±0.35	72.51±0.74	N/A	86.32±0.91	N/A	N/A	N/A	N/A
TOGL [4]	78.59±0.47	72.22±0.79	N/A	90.49±0.76	N/A	N/A	N/A	N/A
DDM [20]	73.93±0.77	75.47±0.50	N/A	91.51±1.45	83.64±0.80	79.88±0.34	62.11±2.35	65.09±0.97
ZS-DM (Ours)	**81.80±0.09**	**76.08±0.19**	**78.93±0.32**	**91.68±0.34**	**86.20±0.12**	**81.73±0.86**	**64.02±1.00**	**66.76±0.24**

Experimental Results: Molecular Property Prediction (ICLR'25)

Performance comparison on ogbg-molhiv (ROC-AUC)

Model	**ogbg-molhiv**
GraphCL	65.18±2.53
TOGL	62.63±2.39
ZS-DM (ours)	**67.52±3.00**

Experimental Results: Spatio-Temporal Forecasting (ICLR'25)

Performance comparison on PeMSD3 and PeMSD8

Probabilistic Models	**PeMSD3**			**PeMSD8**		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
Latent ODE [10]	17.25	28.33	<u>17.71</u>	26.05	39.50	17.20
DeepAR [11]	17.44	28.51	18.02	21.56	33.37	14.15
CSDI [16]	18.92	30.41	19.56	32.11	47.40	18.88
TimeGrad [9]	17.93	29.81	19.33	24.46	38.06	17.03
MC Dropout [18]	<u>17.25</u>	<u>27.85</u>	17.79	19.01	29.35	13.10
DiffSTG [17]	17.79	28.74	18.12	<u>18.60</u>	<u>28.20</u>	<u>11.94</u>
ZS-DM (ours)	**16.57**	**26.46**	**16.25**	**17.59**	**26.09**	**10.29**

Experimental Results: Ablation Study (ICLR'25)

Ablation study of different zigzag-based topological features

Architecture	PROTEINS	MUTAG	PTC_MR
ZS-DM	**76.08±0.19**	**91.68±0.34**	**64.02±1.00**
ZPI-DM	75.83±0.41	89.28±0.90	63.06±1.32
ZFC-DM	75.80±0.49	90.74±0.31	62.43±1.43

Experimental Results: Robustness (ICLR'25)

Robustness study on graph classification for MUTAG under additive Gaussian noise

Model	**Clean**	**1% noise**	**5% noise**
DDM	91.51±1.45	89.68±0.70	86.41±0.76
ZS-DM (ours)	**91.68±0.34**	**91.02±0.83**	**89.76±0.36**

Experimental Results: ZS vs. PH (ICLR'25)

Performance comparison between ZS and traditional persistence on graph classification

Data	ZS	Traditional Persistence
MUTAG	**91.68±0.34**	86.00±0.83
BZR	**86.20±0.12**	83.94±0.37

Experimental Results: Computational Costs (ICLR'25)

Average time taken comparison between ZS-DM and baseline methods

Data	**MUTAG**	**PTC_MR**
DDM	1.73 s	3.41 s
ZS-DM (ours)	2.55 s	3.79 s

Discussion and Future Work

- Zigzag Spaghetti is a tractable time-aware topological summary for *simultaneous assessment* of the topological characteristics of multiple graphs at *all* resolution scales
- Zigzag Spaghetti is more computationally efficient than other time-aware PH summaries without sacrifice in performance
- Zigzag Spaghetti offers a pathway for topological uncertainty quantification.
- Zigzag Spaghetti allows for systematic learning of time-aware topological signatures which are essential for diffusion-based prediction and classification tasks of biomolecular properties within generative diffusion models – outperforming SOTAs in both accuracy and variability
- Zigzag Spaghetti is way bigger than time-aware! Zigzag is based on the diamond principle and can be applied to mix-and-match objects without time index – drug-protein-disease and drug-drug interactions.

[1] Bijaya Adhikari, Yao Zhang, Naren Ramakrishnan, and B Aditya Prakash. Sub2vec: Feature learning for subgraphs. In *PAKDD*, pages 170–182. Springer, 2018.

[2] Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Lio, Guido F Montufar, and Michael Bronstein. Weisfeiler and lehman go cellular: CW networks. In *NeurIPS*, volume 34, pages 2625–2640, 2021.

[3] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, pages 855–864, 2016.

[4] Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. Topological graph neural networks. In *ICLR*, 2022.

[5] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. GraphMAE: Self-supervised masked graph autoencoders. In *SIGKDD*, pages 594–604, 2022.

[6] Sihang Li, Xiang Wang, An Zhang, Yingxin Wu, Xiangnan He, and Tat-Seng Chua. Let invariant rationale discovery inspire graph contrastive learning. In *ICML*, 2022.

[7] Lu Lin, Jinghui Chen, and Hongning Wang. Spectrum guided topology augmentation for graph contrastive learning. In *NeurIPS 2022 Workshop: New Frontiers in Graph Learning*, 2022.

[8] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*, 2017.

[9] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *ICML*, pages 8857–8868. PMLR, 2021.

[10] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *NeurIPS*, volume 32, 2019.

[11] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[12] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten M Borgwardt. Efficient graphlet kernels for large graph comparison. In *AISTATS*, 2009.

[13] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *JMLR*, 12(9), 2011.

[14] Fan-Yun Sun, Jordan Hoffman, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*, 2019.

[15] Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. Adversarial graph augmentation to improve graph contrastive learning. *NeurIPS*, 34:15920–15933, 2021.

[16] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *NeurIPS*, volume 34, pages 24804–24816, 2021.

[17] Haomin Wen, Youfang Lin, Yutong Xia, Huaiyu Wan, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. DIFFSTG: Probabilistic spatio-temporal graph forecasting with denoising diffusion models. In *SIGSPATIAL*, pages 1–12, 2023.

[18] Dongxia Wu, Liyao Gao, Matteo Chinazzi, Xinyue Xiong, Alessandro Vespignani, Yi-An Ma, and Rose Yu. Quantifying uncertainty in deep spatiotemporal forecasting. In *SIGKDD*, pages 1841–1851, 2021.

[19] Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *SIGKDD*, pages 1365–1374, 2015.

[20] Run Yang, Yuling Yang, Fan Zhou, and Qiang Sun. Directional diffusion models for graph representation learning. In *NeurIPS*, volume 36, 2024.

[21] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *NeurIPS*, volume 33, pages 5812–5823, 2020.