DspGNN

Bringing **Spectral Design** to Discrete Time **Dynamic Graph Neural Networks** for **Edge Regression**

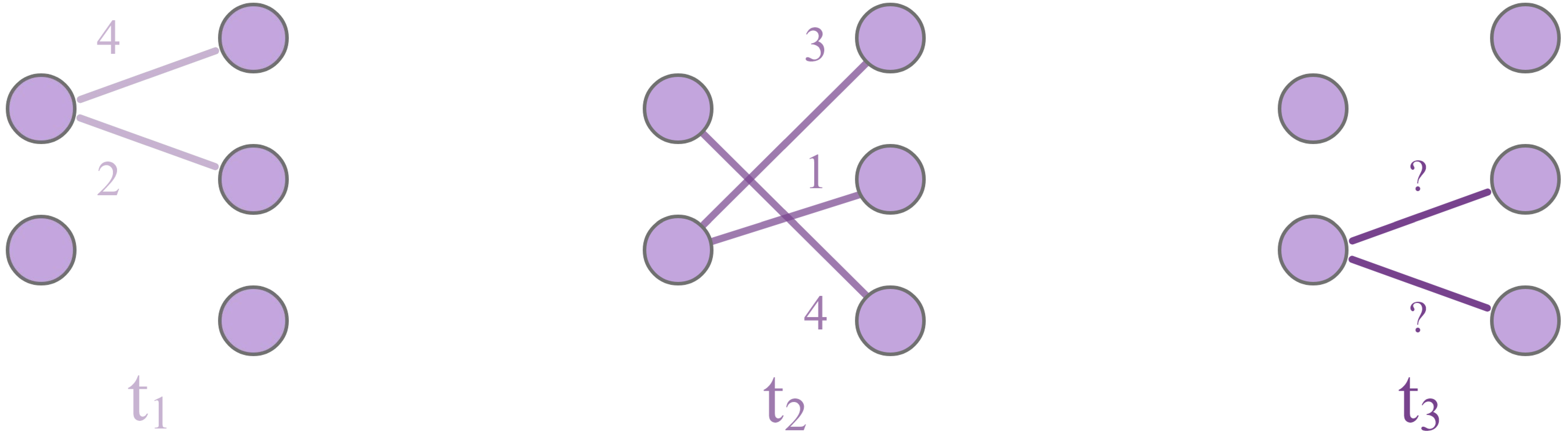
Leshanshui Yang [†, §, ‡], Clément Chatelain [†, #], Sébastien Adam [†, ‡]

anr agence nationale de la recherche · L-LISA Laboratoire LITIS Saagie · litis · Saagie · Université de Rouen Normandie · INSA Institut National des Sciences Appliquées Rouen

† § ‡ #

L. Yang, C. Chatelain, and S. Adam, « DspGNN: Bringing Spectral Design to DTDGNNs for Edge Regression » in Temporal Graph Learning Workshop@NeurIPS, 2023.

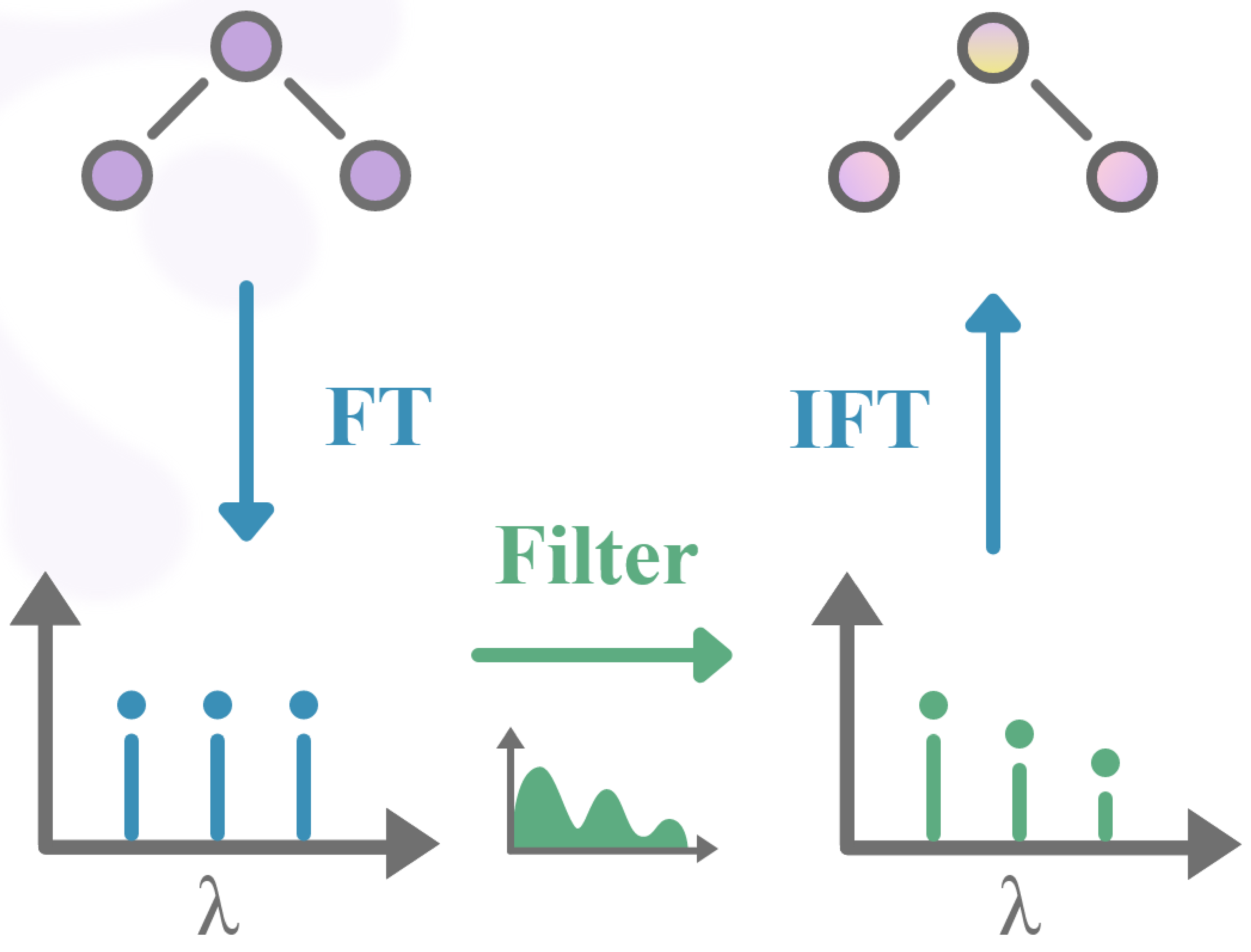
Discrete Time Dynamic Graph Edge Regression



Given

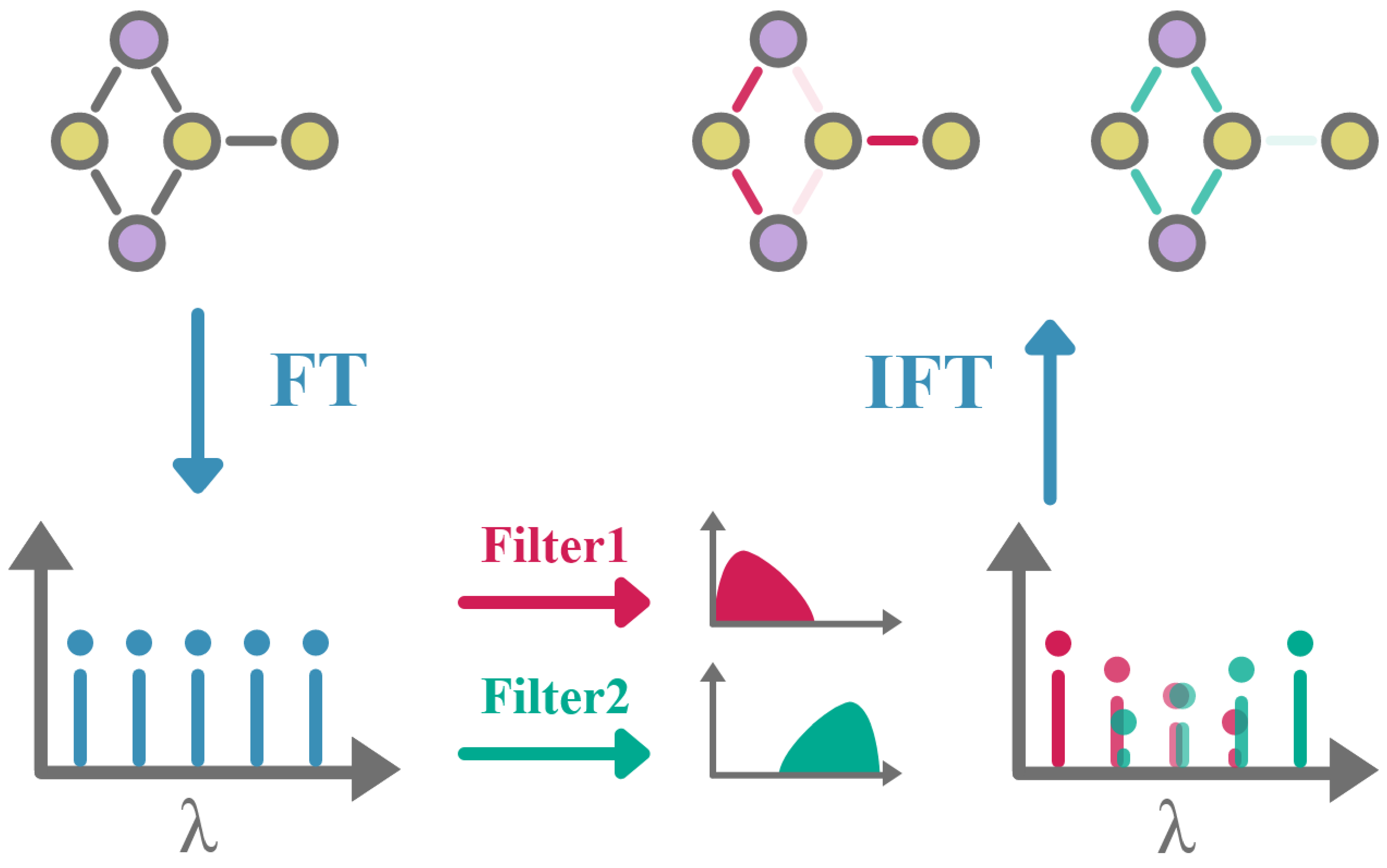
Predict

Spectral-Designed Graph Convolution



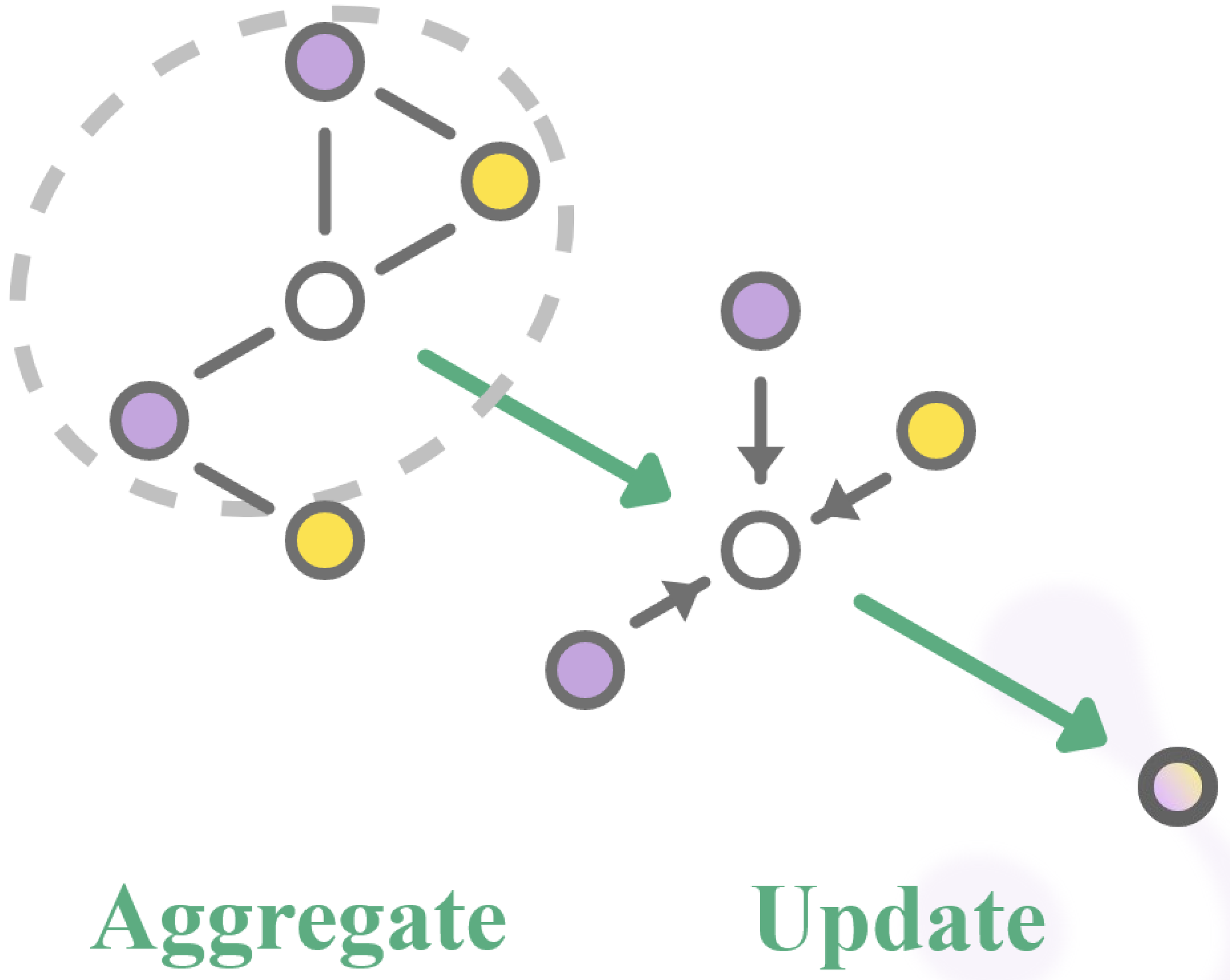
Spectral Convolution

$$\mathbf{H}' = \mathbf{U}\mathbf{g}(\boldsymbol{\Lambda})\mathbf{U}^{\mathbf{T}}\mathbf{H}$$



Spectral-Designed Convolution

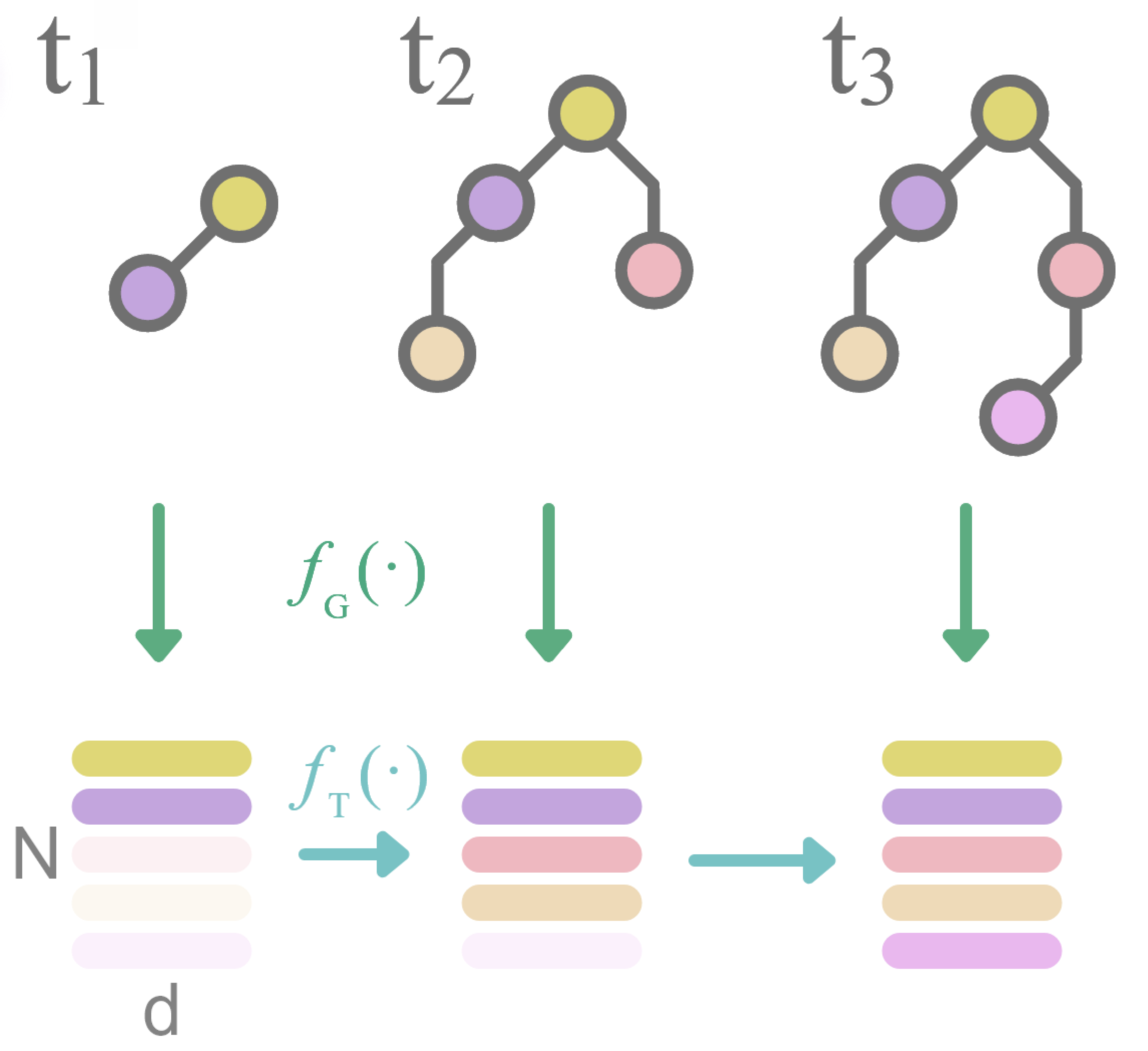
$$\mathbf{H}' = \sigma\left(\sum_{\mathbf{s}} \mathbf{C}^{(\mathbf{s})}\mathbf{H}\mathbf{W}^{(\mathbf{s})}\right), \quad \mathbf{C}^{\mathbf{s}} = \mathbf{U}\boldsymbol{\Phi}^{\mathbf{s}}(\boldsymbol{\Lambda})\mathbf{U}^{\mathbf{T}}$$



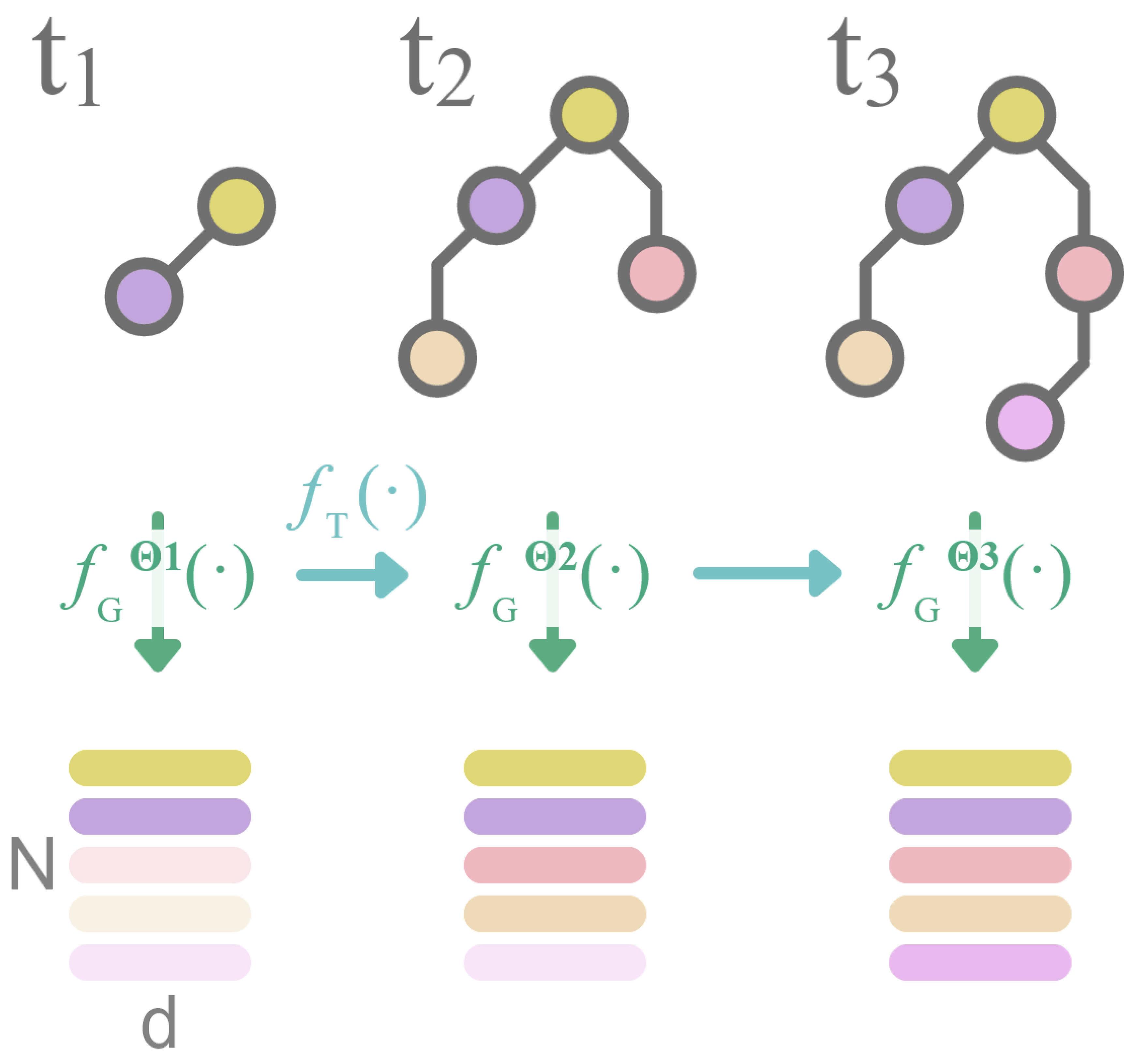
Spatial Convolution

$$\mathbf{H}' = \sigma\left(\sum_{\mathbf{s}} \mathbf{C}^{(\mathbf{s})}\mathbf{H}\mathbf{W}^{(\mathbf{s})}\right)$$

Discrete Time Dynamic Graph Neural Networks

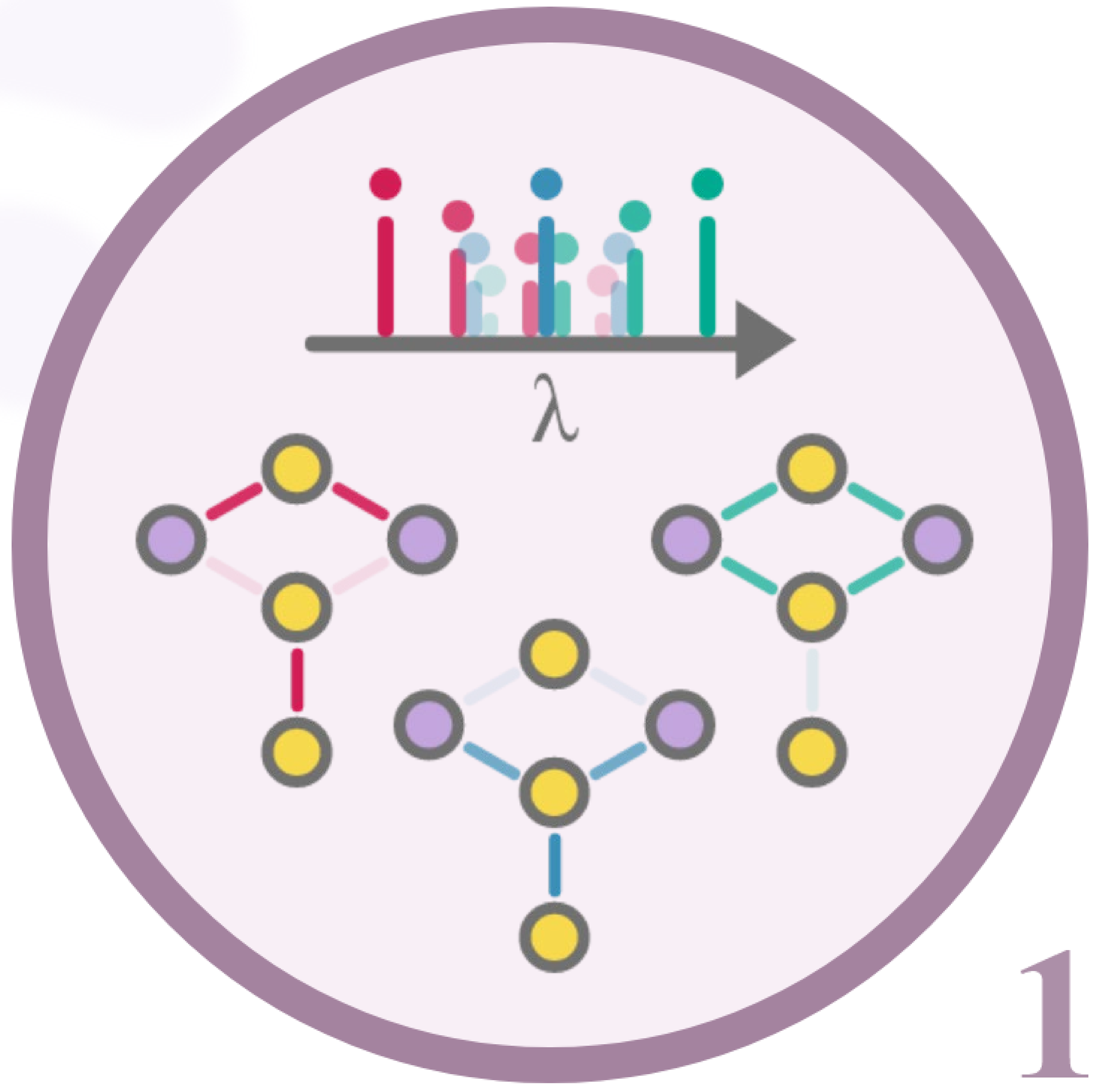


Sequentially Enc(H)

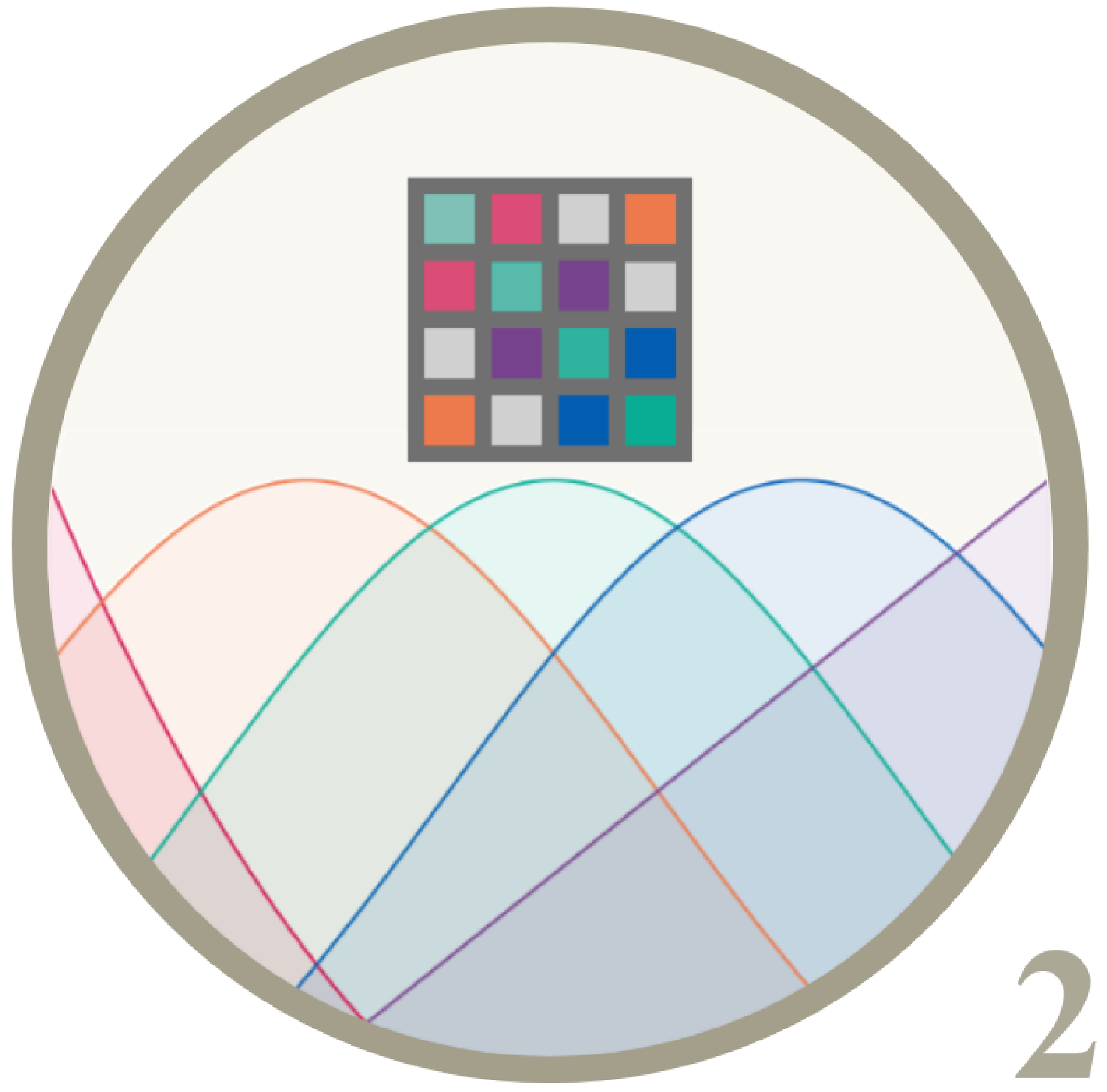


Sequentially Enc(Θ)

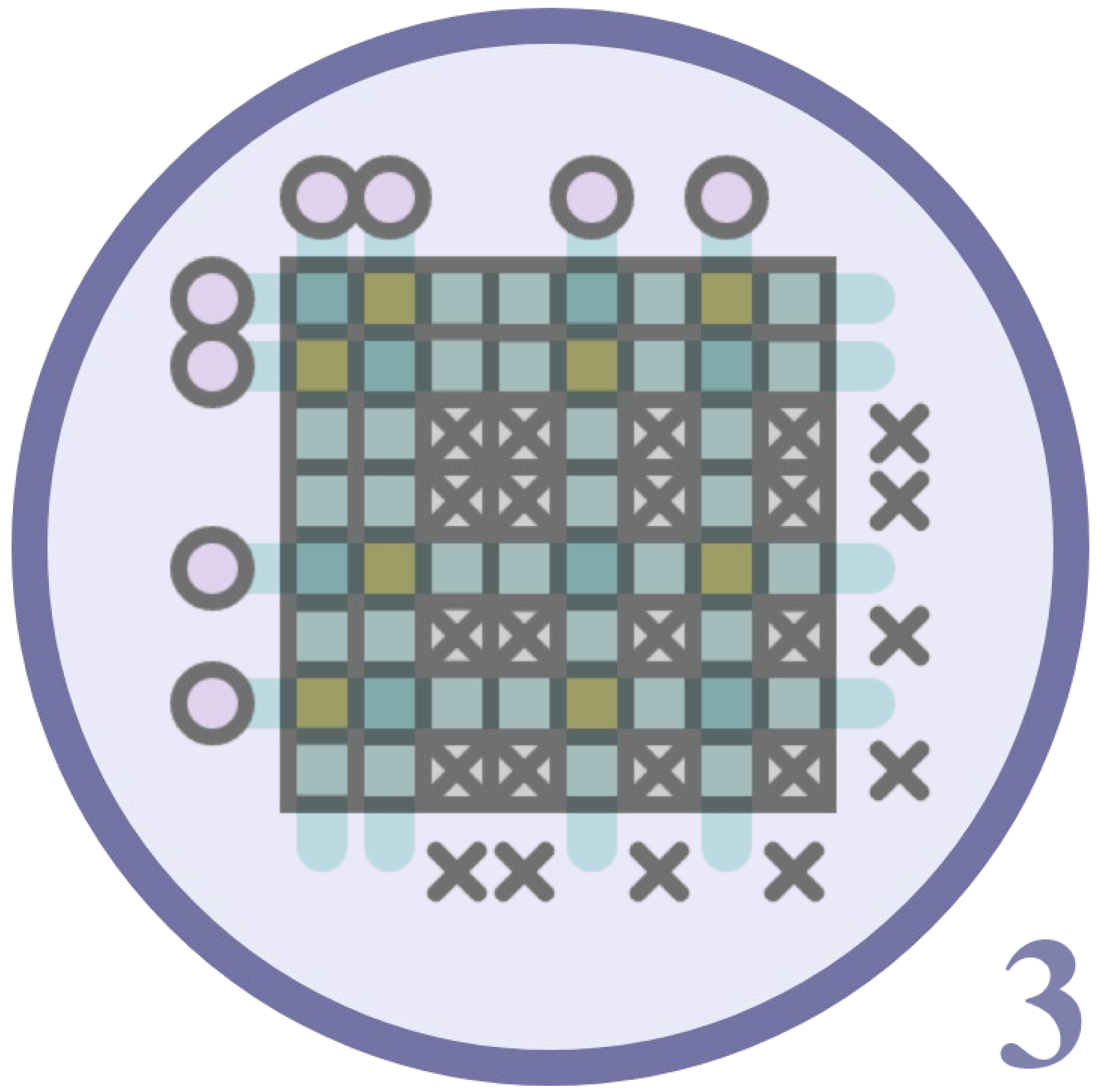
Motivation



DSGCN in DTDGNNs

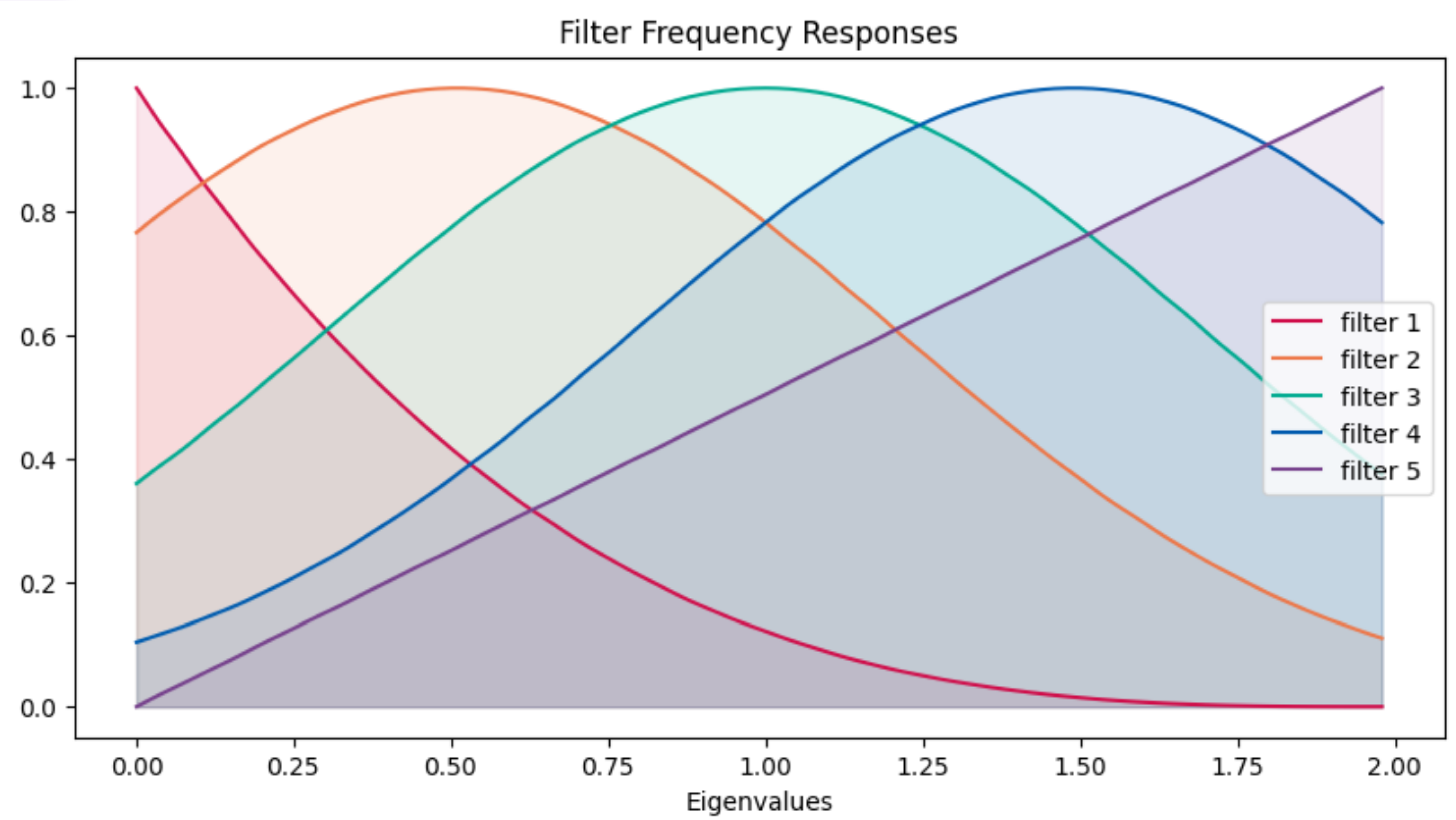


Combine filter outputs



Faster eigendecomposition

Spectral-Designed Filters



$$F_1(\lambda) = \left(1 - \frac{\lambda}{\lambda_{\text{max}}}\right)^3, \qquad \text{(Low-pass)}$$

$$F_2(\lambda) = e^{-((\lambda - \lambda_{c1})^2) \cdot \gamma}, \qquad \text{(Band-pass 1)}$$

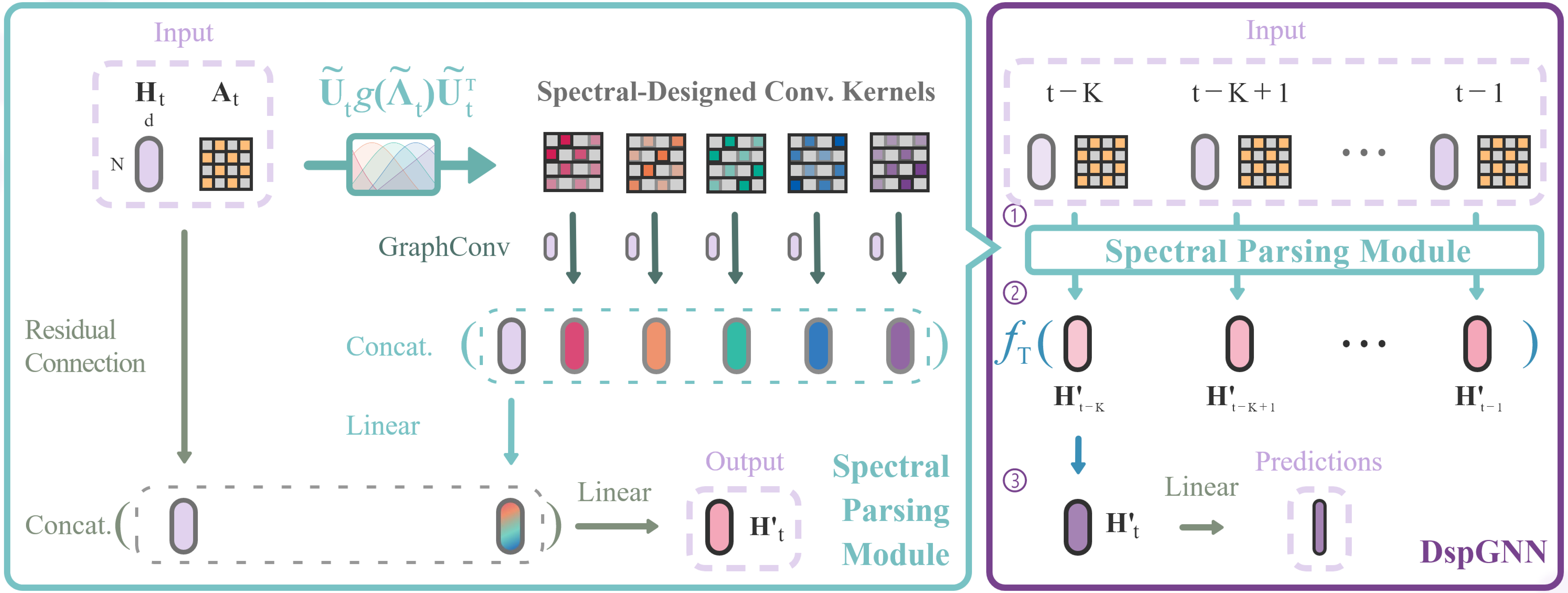
$$F_3(\lambda) = e^{-((\lambda - \lambda_{c2})^2) \cdot \gamma}, \qquad \text{(Band-pass 2)}$$

$$F_4(\lambda) = e^{-((\lambda - \lambda_{c3})^2) \cdot \gamma}, \qquad \text{(Band-pass 3)}$$

$$F_5(\lambda) = \frac{\lambda}{\lambda_{\text{max}}}, \qquad \text{(High-pass)}$$

DspGNN (Spectral Parsing Module)

DspGNN (Overall Architecture)



① Encode snapshots SP Module ② Encode with a temporal encoder ③ Compute the predictions

Active Node Mapping

Without Active Node Mapping $O(N^3)$

With Active Node Mapping $O(N_t^3)$

Experiments

Name	# Nodes	# Average active nodes per snapshot	# Edges	# Time steps	Time step split Train / Valid / Test
Bitcoin-Alpha	3,783	106	24,173	137	95 / 14 / 28
Bitcoin-OTC	5,881	148	35,588	136	95 / 13 / 28
MovieLens-100K	9,811	740	100,836	90	63 / 9 / 18

Bitcoin: level of trust, -10~+10 MovieLens-100K: movie rating, 0.5~5.0

	Bitcoin-Alpha		Bitcoin-OTC		MovieLens-100K	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
CoEvoGNN*	*0.1171* (0.0097)	*0.0879* (0.0100)	0.1638 (0.0066)	0.1145 (0.0106)	0.2381 (0.0058)	*0.1834* (0.0048)
CoEvoGNN	0.1231 (0.0033)	0.0928 (0.0030)	0.1671 (0.0073)	0.1176 (0.0064)	*0.2372* (0.0059)	0.1894 (0.0067)
EvolveGCN	0.1173 (0.0187)	0.0903 (0.0216)	*0.1556* (0.0049)	*0.1025* (0.0067)	0.2404 (0.0134)	0.1944 (0.0160)
DspGNN(Ours)	**0.0968** (0.0002)	**0.0637** (0.0006)	**0.1471** (0.0009)	**0.0831** (0.0019)	**0.2293** (0.0000)	**0.1812** (0.0000)

Edge regression performance: MEAN and STD of the normalized scores for the five seeds

Computing cost comparison

		W/O Active Node Mapping		With Active Node Mapping
Max space complexity (Theo.)	Bitcoin-Alpha		1×10^7	3×10^5
	Bitcoin-OTC		3×10^7	3×10^5
	MovieLens-100K		1×10^8	6×10^6
Time complexity (Theo.)	Bitcoin-Alpha		7×10^{12}	6×10^8
	Bitcoin-OTC		3×10^{13}	1×10^9
	MovieLens-100K		9×10^{13}	1×10^{11}
Real time consumed (seconds)	Library	Numpy	Scipy	Numpy
	Bitcoin-Alpha	1826.31	225.69	0.31
	Bitcoin-OTC	7203.62	592.23	0.55
	MovieLens-100K	11150.19	1159.91	12.83

Acknowledgments

NEURAL INFORMATION PROCESSING SYSTEMS

TGL

anr agence nationale de la recherche

L-LISA Laboratoire LITIS Saagie

ANR Labcom Lisa ANR-20-LCV1-0009

Reference

DSGCN
[1] Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gauzere, Sebastien Adam, and Paul Honeine. Bridging the gap between spectral and spatial domains in graph neural networks. arXiv preprint arXiv:2003.11702, 2020.
[2] Muhammet Balcilar, Renton Guillaume, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Analyzing the expressive power of graph neural networks in a spectral perspective. In Proceedings of the International Conference on Learning Representations (ICLR), 2021.

CoEvoGNN
[3] Daheng Wang, Zhihan Zhang, Yihong Ma, Tong Zhao, Tianwen Jiang, Nitesh Chawla, and Meng Jiang. Modeling co-evolution of attributed and structural information in graph sequence. IEEE Transactions on Knowledge and Data Engineering, 2021.

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

EvolveGCN
[5] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kaneza-shi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pages 5363–5370, 2020.

Bitcoin-Alpha, Bitcoin OTC
[6] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. In Data Mining (ICDM), 2016 IEEE 16th International Conference on, pages 221–230. IEEE, 2016.

MovieLens-100K
[7] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis), 5(4):1–19,