

Graph Representation Learning for Multimodal Data- Challenges and Innovative Methods

Maysam Behmanesh

behmanesh@lix.Polytechnique.fr



April 4th - 2024

Outlines

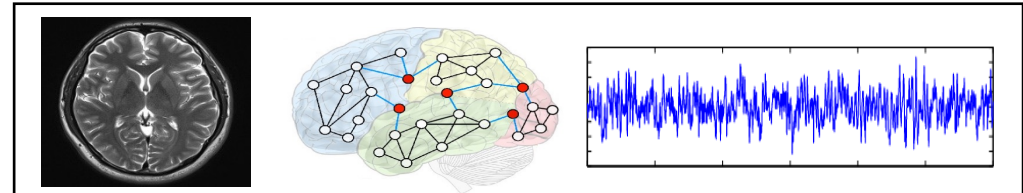
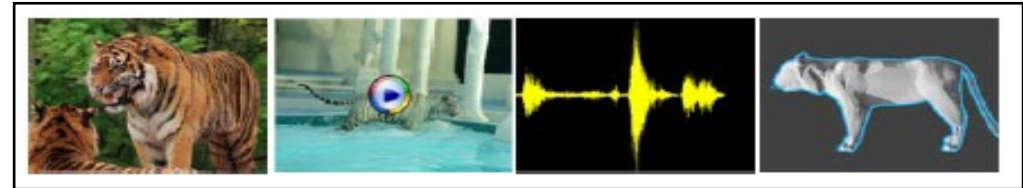
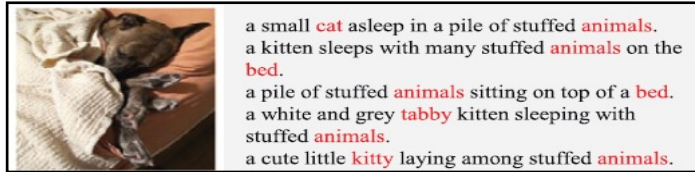
1. Benefits of multimodal data
2. Geometry processing
3. Graph multimodal learning
4. Challenges and innovative methods
5. Ongoing projects

Motivation

1. Benefits of multimodal data

Growth of *diverse* data that incorporate information from multiple sources or modalities

Autonomous Driving, Multimodal Machine Translation, Emotion Recognition, Image Captioning, Visual Question Answering (VQA)



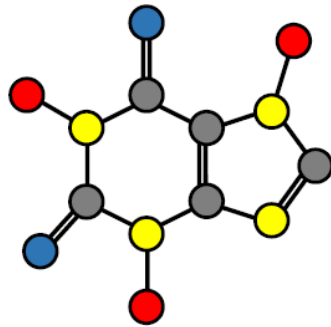
Motivation

2. Geometry is Everywhere!

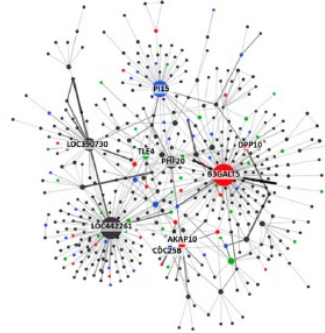
Growth of *diverse* geometry based data: Social networks, Molecules, Interaction networks, Bio-medical imaging, 3D shape analysis



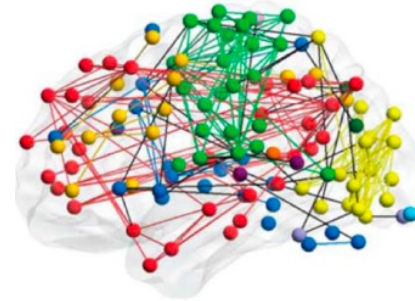
Social networks



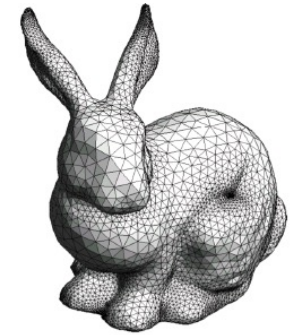
Molecules



Interaction networks



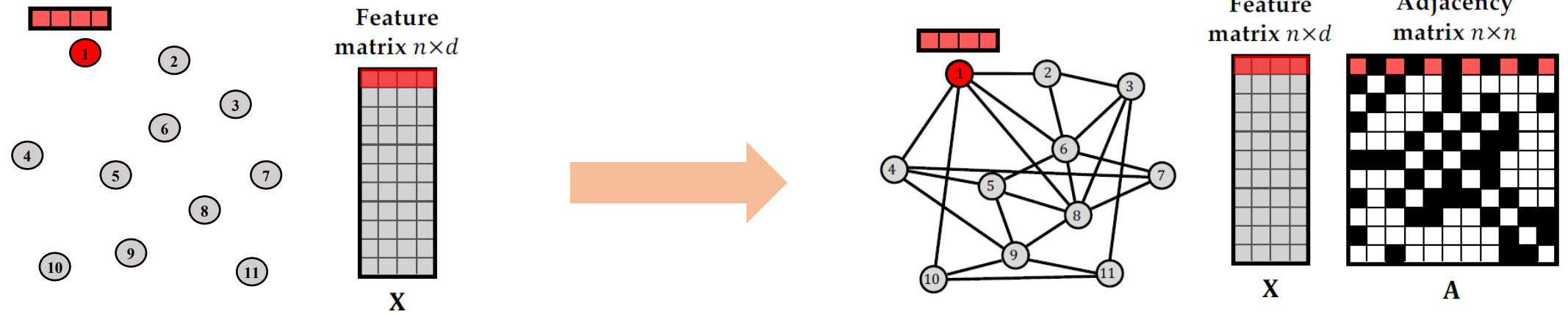
Functional networks



Meshes

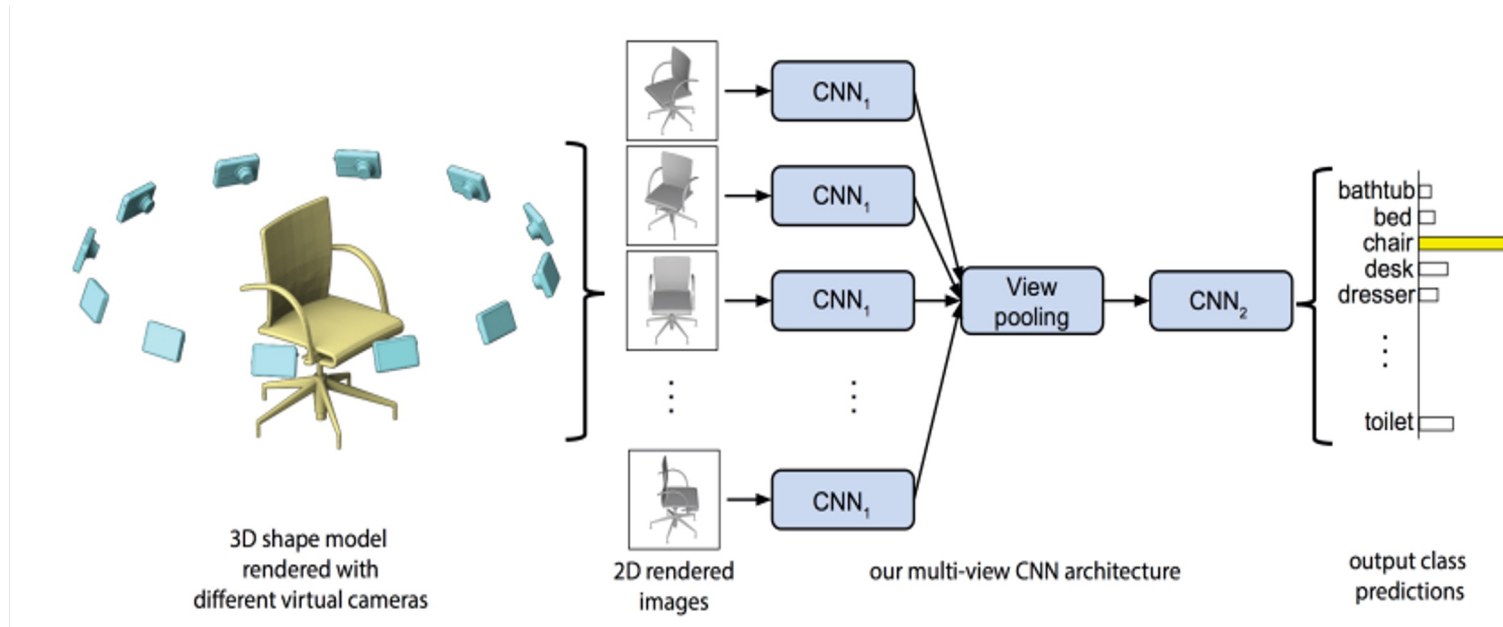
3. Implicit graphs

Inject geometric information into point cloud to form an *implicit graph*



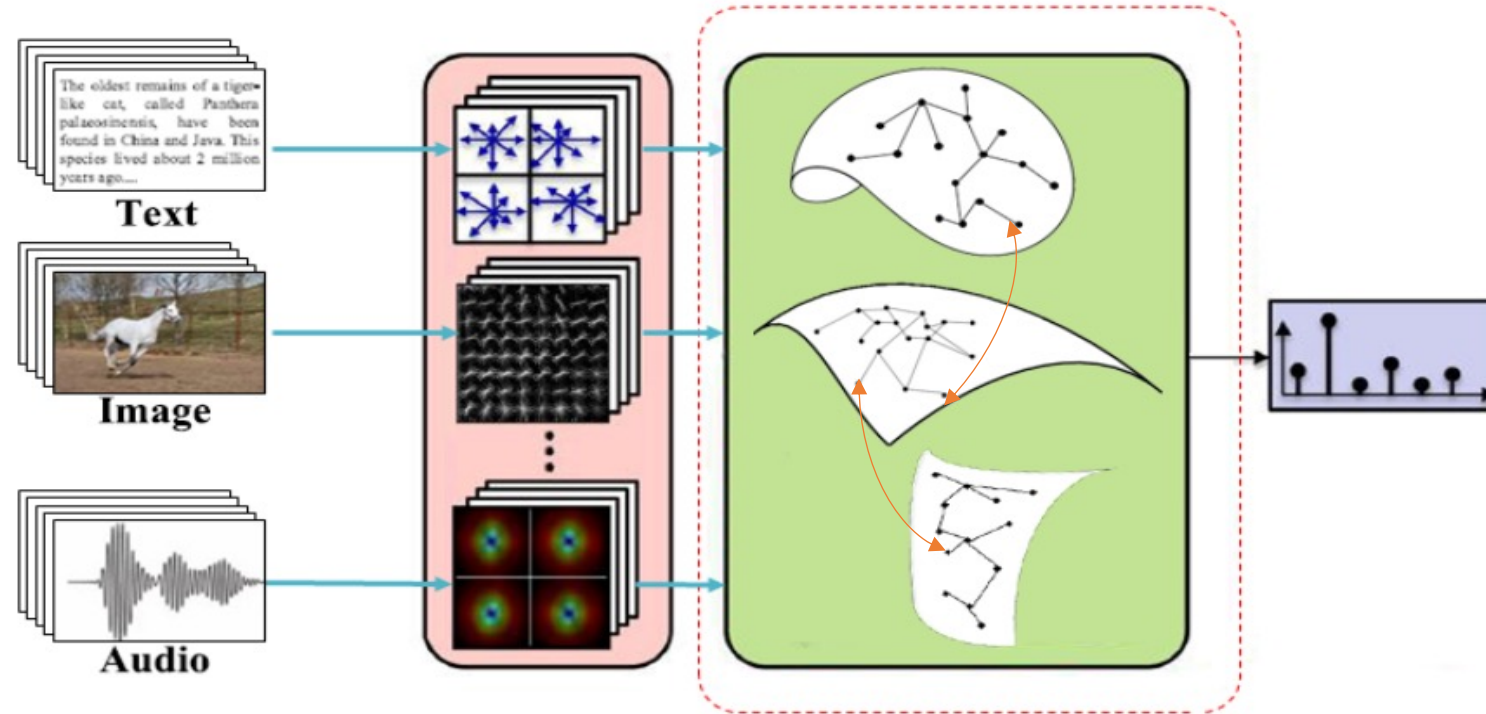
Major limitations

- Most multimodal methods **are constrained to the particular cases.**
- Limited to *prior knowledge* and *homogeneous data*, not useful in generic tasks!



Research direction

Geometric Multimodal Learning in a practical scenario



Data fusion: how to integrate data from heterogeneous modalities?

Translation: how to find correspondences among data in different modalities?

Multimodal Multi-scaled Graph Wavelet Convolutional Network (M-GWCN)

Geometric Multimodal Deep Learning with Multi-Scaled Graph Wavelet Convolutional Network

Maysam Behmanesh, Peyman Adibi, Mohammad Saeed Ehsani, and Jocelyn Chaussoot, *Fellow, IEEE*

Abstract—Multimodal data provide complementary information of a natural phenomenon by integrating data from various domains with very different statistical properties. Capturing the intra-modality and cross-modality information of multimodal data is the essential capability of multimodal learning methods. The geometry-aware data analysis approaches provide these capabilities by implicitly representing data in various modalities based on their geometric underlying structures. Also, in many applications, data are explicitly defined on an intrinsic geometric structure. Generalizing deep learning methods to the non-Euclidean domains is an emerging research field, which has recently been investigated in many studies. Most of those popular methods are developed for unimodal data. In this paper, a multimodal multi-scaled graph wavelet convolutional network (M-GWCN) is proposed as an end-to-end network. M-GWCN simultaneously finds intra-modality representation by applying the multiscale graph wavelet transform to provide helpful localization properties in the graph domain of each modality, and cross-modality representation by learning permutations that encode correlations among various modalities. M-GWCN is not limited to either the homogeneous modalities with the same number of data, or any prior knowledge indicating correspondences between modalities. Several semi-supervised node classification experiments have been conducted on three popular unimodal explicit graph-based datasets and five multimodal implicit ones. The experimental results indicate the superiority and effectiveness of the proposed methods compared with both spectral graph domain convolutional neural networks and state-of-the-art multimodal methods.

Index Terms—Geometric deep learning, Graph convolution neural networks, Graph wavelet transform, Multimodal learning, Spectral approaches

I. INTRODUCTION

by discovering the hidden intra-modality and cross-modality correlations.

However, although recent multimodal models have been focused on Euclidean data, there are two major situations in which data should be processed in non-Euclidean domains. First, in the cases in which data in various modalities are implicitly represented based on their geometric structures. Second, when data are generated in non-Euclidean geometric domains, and inherently defined for example as a graph. These applications represent complex relationships and interdependencies among objects [2], including social networks, citation networks, networks of the spread of epidemic diseases, e-commerce networks, brain's neuronal networks, biological regulatory networks, and so on.

With the emergence of geometric structural data in real-world applications, many works have investigated generalizing deep learning methods to the non-Euclidean domains [2], [3]. As the most popular challenges for the graphs domain data, graph neural networks (GNNs) perform filtering operations directly on the graph via the graph weights [3] and graph convolutional networks (GCNs) learn the local meaningful stationary properties of the input signals through specifically designed convolution operator on graphs [4]. However, complex geometric structures in graphs can be encoded with more powerful mathematical tools in many spatial or spectral graph-based methods [5]. Nevertheless, most of these popular methods are developed for unimodal data and have difficulties coping with multimodal problems.

One of the remarkable deficiencies of the previous multimodal data analysis methods is their limitation to com-



Maysam Behmanesh



Peyman Adibi



Saeed Ehsani



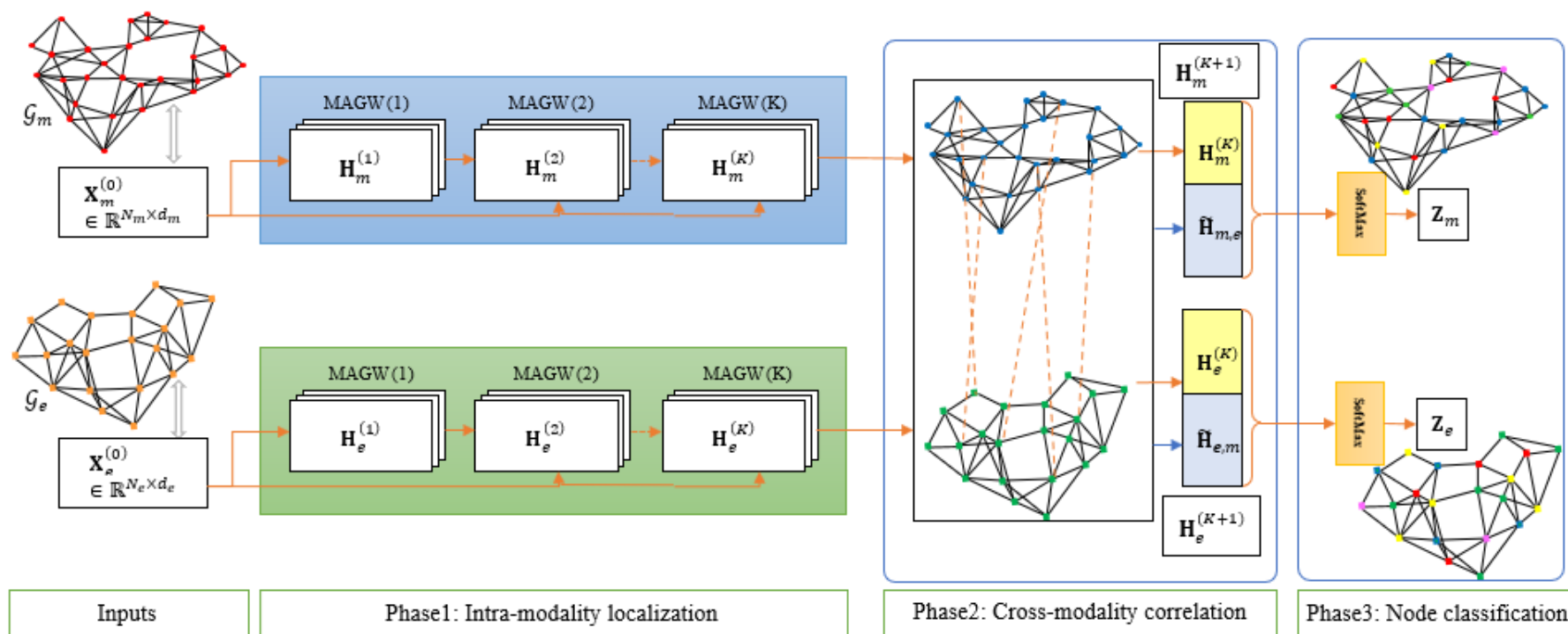
Jocelyn Chaussoot

<https://github.com/maysambehmanesh/GWCN>

Geometric Multimodal Deep Learning with Multi-Scaled Graph Wavelet Convolutional Network,” M. Behmanesh, P. Adibi, M. S. Ehsani, and J. Chaussoot, IEEE TNNLS, 2022

Overall objectives:

- 1- Feature learning in each modality by *exploring various localities*
- 2- Take advantages of complementary information provided by different modalities



Challenges:

- *Intra-modality representation* in the graph domain of each modality
- *Cross-modality correlations* among various modalities

Intra-modality representation:

Graph Wavelet Convolution Network (GWCN)

From GFT to GWT:

Replacing eigen basis \mathbf{U} with wavelet basis Ψ_s

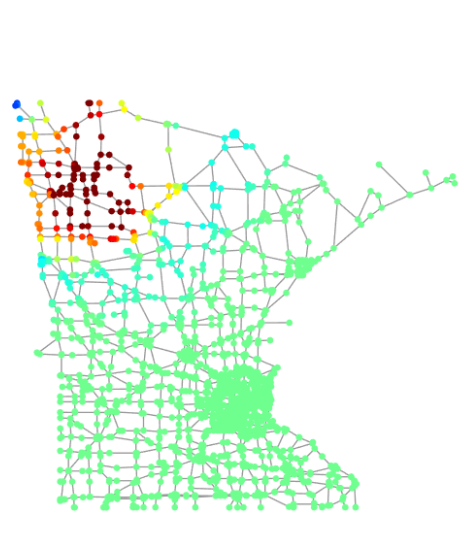
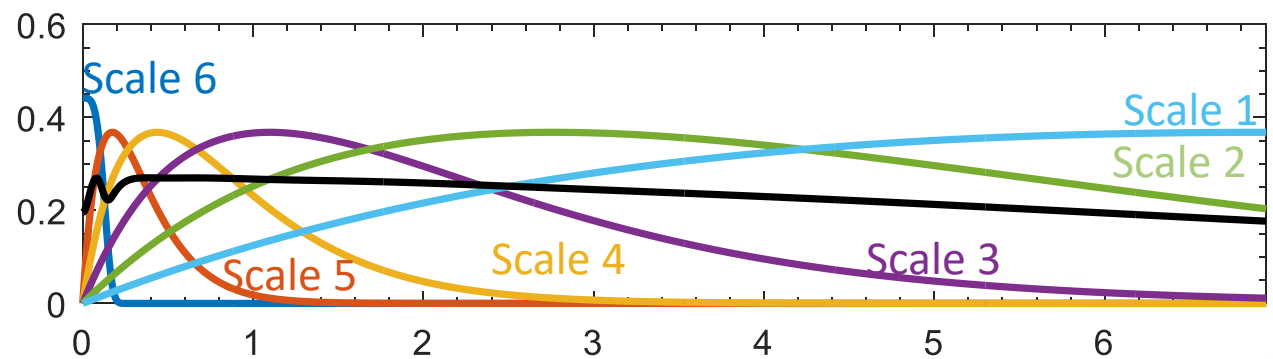
$$\mathbf{f} * \mathbf{g} = \mathbf{U} \cdot \mathbf{g}_\theta(\mathbf{U}^T \mathbf{f})$$

$$\mathbf{f} * \mathbf{g} = \Psi_s \cdot \mathbf{g}_\theta(\Psi_s^{-1} \mathbf{f})$$

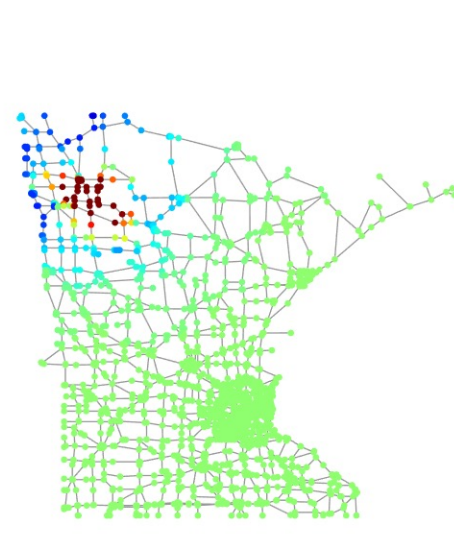
$$\mathbf{H}^{(k+1)} = \sigma(\mathbf{U} \mathbf{W}^{(k)} \mathbf{U}^T \mathbf{H}^k)$$

$$\mathbf{H}_s^{(k+1)} = \sigma(\Psi_s \theta_s \Psi_s^{-1} \mathbf{H}_s^{(k)} \mathbf{W}_s^{(k)})$$

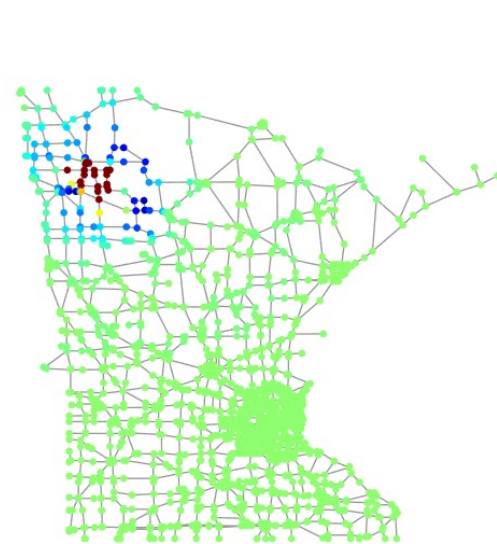
Advantages of wavelet basis



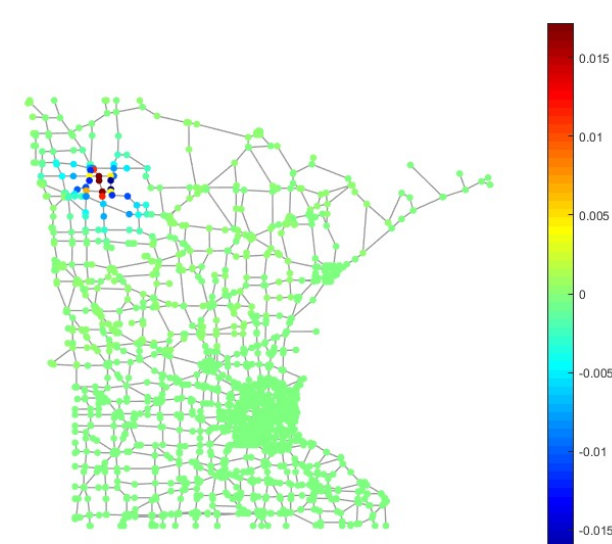
Scale 1



Scale 2



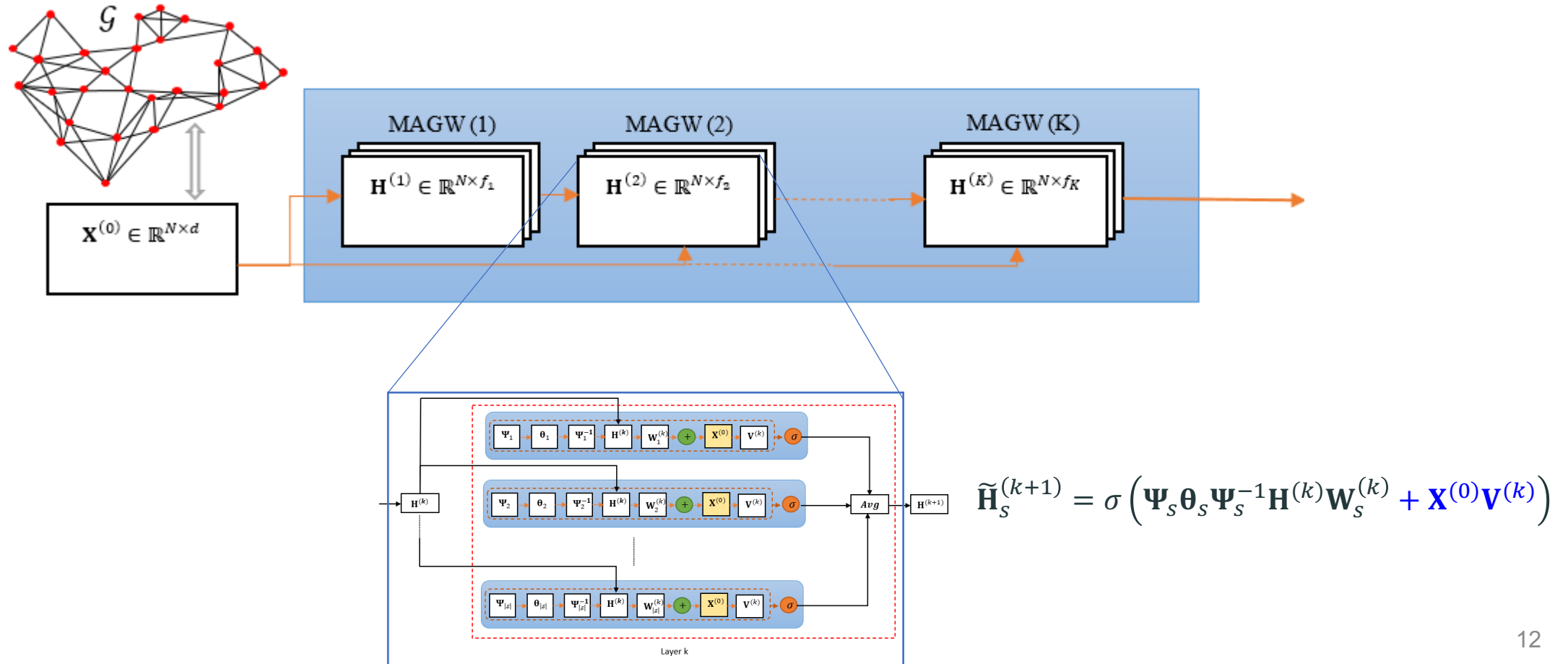
Scale 3



Scale 4

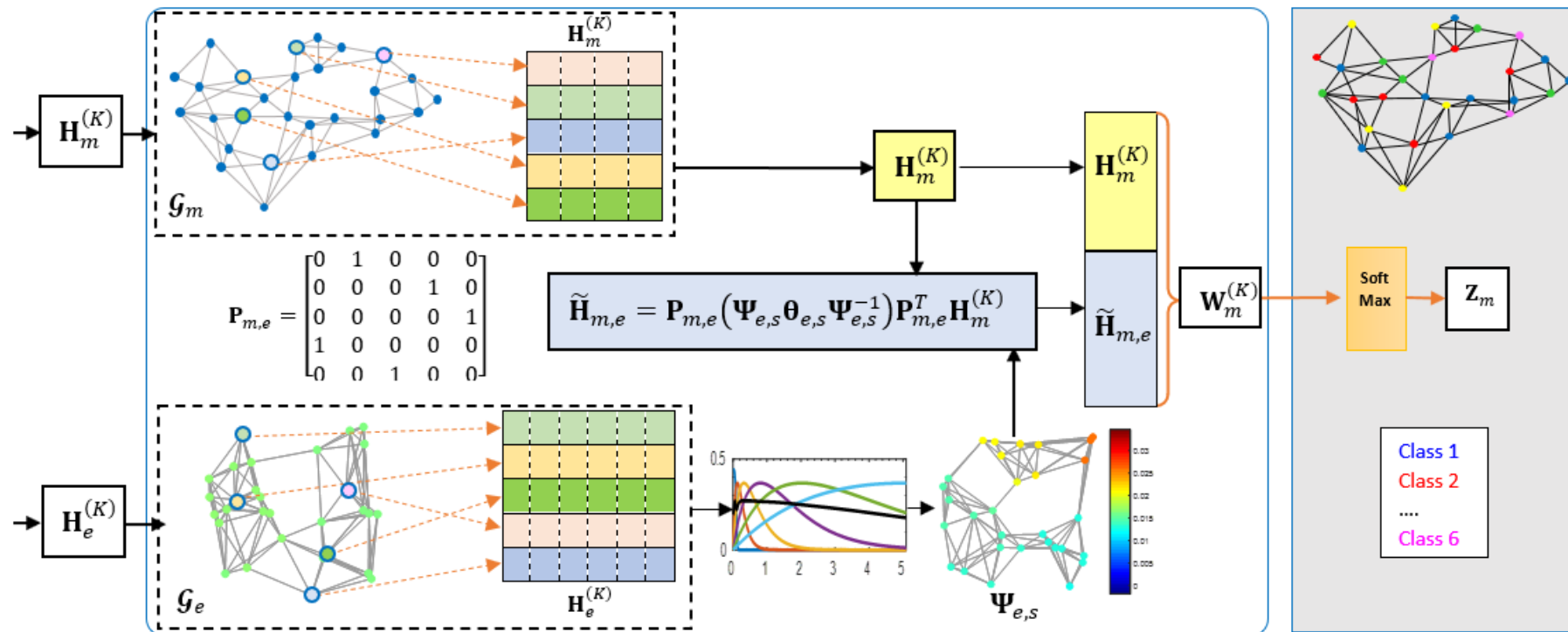
Phase 1: Intra-modality localization

Applying Graph Wavelet Convolution with $|\mathcal{S}|$ different scales



Phase 2: Cross-modality correlations




- 1- Feature embedding of each modality based on the graph wavelet of the other one
- 2- Exploring the point-wise correspondences by learning a permutation matrix



Results: Multimodal Implicit Graph-Based Data

Multimodal

Method	Caltech	NUS
CD (pos)	76.9±1.1	81.8±0.8
CD (pos+neg)	73.4±0.8	80.3±0.4
SCSMM	-	83.9±2.42
m-LSJD	84.1±1.4	83.2±1.3
m ² -LSJD	88.5±1.6	87.2±1.1
M ² CPC-u	84.8±0.7	86.4±0.3
M-GWCN	90.6±0.4	89.2±0.8

visual modality			
	A	B	C
textual modality	'water' 'cat' 'sea' 'washington' 'tiger'	'nature' 'water' 'explore' 'sea' 'fish'	'red' 'macro' 'flower' 'flowers' 'rose'
	D	E	F

Multi-view

Method	Caltech101-7	Caltech101-20	MNIST
MLDA	92.29±8e-3	76.59±12e-3	92.84±5e-3
MLDA-m	89.78±10e-3	73.77±114e-3	93.09±8e-3
MULDA	92.65±8e-3	82.20±11e-3	95.23±5e-3
MULDA-m	92.59±10e-3	82.17±6e-3	95.12±4e-3
MvMDA	92.65±8e-3	80.50±13e-3	93.78±9e-3
OGMA	95.01±5e-3	86.00±10e-3	96.09±6e-3
OMLDA	94.98±5e-3	86.85±10e-3	95.71±6e-3
OMvMDA	94.71±7e-3	82.28±10e-3	95.99±6e-3
M ² CPC-p	94.83±1.1	86.44±1.1	-
MV-GWCN-1	95.25±1.3	87.83±0.8	96.45±1.4
MV-GWCN-2	96.23±0.7	88.46±1.1	97.21±0.7

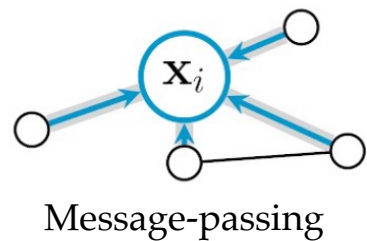
Dataset	Type	No. modalities	No. data samples	No. classes
Caltech	multimodal	2	1474	7
NUS	multimodal	2	6000	7
Caltech-101-7	multi-view	6	1474	7
Caltech-101-20	multi-view	6	2386	20
MNIST	multi-view	6	2000	10

Major challenges...

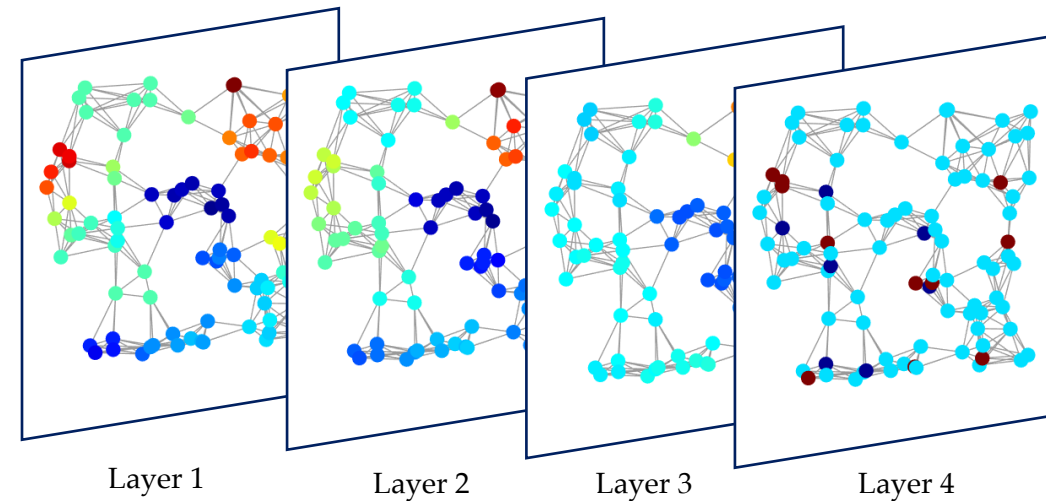
1. The GWCN model is prone to *oversmoothing*
2. Limited to *data-rich applications*, not useful in generic tasks
3. Optimal Transport (OT)-based loss, can be *computationally expensive* and may not scale well to *large-scale graphs*

Challenge

- Message-passing based approaches are prone to oversmoothing
- Most GNNs are **constrained to small scaled graph**



$$\mathbf{h}_i = \phi \left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} \psi(\mathbf{x}_i, \mathbf{x}_j) \right)$$



Our Goals:

- Avoid structural limitations of the message-passing frameworks
- Facilitate information propagation by using the diffusion equation
- Ensure long-distance communication between nodes

TIDE: Time Derivative Diffusion for Deep Learning on Graphs

TIDE: Time Derivative Diffusion for Deep Learning on Graphs

Maysam Behmanesh^{*1} Maximilian Krahn^{*1,2} Maks Ovsjanikov¹

Abstract

A prominent paradigm for graph neural networks is based on the message-passing framework. In this framework, information communication is realized only between neighboring nodes. The challenge of approaches that use this paradigm is to ensure efficient and accurate *long-distance communication* between nodes, as deep convolutional networks are prone to oversmoothing. In this paper, we present a novel method based on time derivative graph diffusion (TIDE) to overcome these structural limitations of the message-passing framework. Our approach allows for optimizing the spatial extent of diffusion across various tasks and network channels, thus enabling medium and long-distance communication efficiently. Furthermore, we show that our architecture design also enables local message-passing and thus inherits from the capabilities of local message-passing approaches. We show that on both widely used graph benchmarks and synthetic mesh and graph datasets, the proposed framework outperforms state-of-the-art methods by a significant margin.⁺

(see, e.g., (Zhou et al., 2020; Wu et al., 2020) for recent surveys), ranging from spectral methods, spatial or convolutional designs, recurrent graph neural networks, or graph auto-encoders as well as many other hybrid techniques. A particularly prominent and widely-used category of approaches is given by the convolutional graph neural networks, and especially those based on message-passing, following the design introduced in (Kipf & Welling, 2017) and extended significantly in many follow-up works, e.g., (Li et al., 2018b; Zhuang & Ma, 2018; Chamberlain et al., 2021b; Thorpe et al., 2021).

The key strengths of convolutional graph neural networks, as introduced in (Kipf & Welling, 2017), include their simplicity and computational efficiency, their ability to be composed with other neural networks as well as their ability to generalize across different graphs (i.e., learning weights that could be applied on unseen graphs). As a result, the original GCN approach (Kipf & Welling, 2017) is still highly effective and is widely used in many applications.

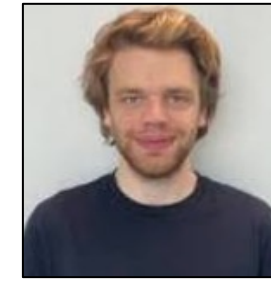
Nevertheless, a prominent limitation of message-passing approaches, such as GCN and related methods is *oversmoothing*, which implies that such networks tend to be difficult to train beyond a small number of layers (Oono & Suzuki, 2019). Furthermore, since typical message-passing operators only ensure communication between nodes within a 1-hop neighborhood, this means that message-passing approaches can hinder *long-distance information propagation*, which can limit their utility in scenarios, where such long-range communication is important.

1. Introduction

Designing efficient and scalable architectures for learning on graphs is a central problem in machine learning with applications in a broad range of disciplines, including data



Maysam Behmanesh



Maximilian Krahn



Maks Ovsjanikov

<https://github.com/maysambehmanesh/TIDE>

“TIDE: Time Derivative Diffusion for Deep Learning on Graphs,” M. Behmanesh, M. Krahn, and M. Ovsjanikov, ICML, 2023

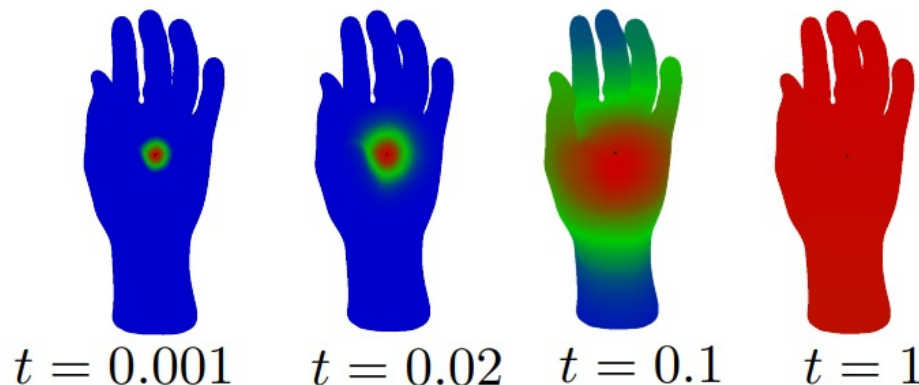
Laplacians and diffusion

In the continuous setting, the diffusion process is described as the solution of *the heat equation*

$$\frac{\partial u}{\partial t} = -\Delta u$$

- ↳ Basic linear PDE
- ↳ defined on surfaces via the Laplace-Beltrami operator Δ
- ↳ implemented & well-studied on many domains

————— diffusion of a point value —————>



$$\begin{aligned}
 u_t &= \mathcal{H}_t(u_0) \\
 &= \exp(-t\Delta) u_0 \\
 \mathcal{H}_t &: \text{Heat operator}
 \end{aligned}$$

Time-derivative diffusion

Main goal:

Combine the *local accuracy* with the *global information propagation* (without oversmoothing)

We propose time-derivative diffusion as a communication mechanism:

$$\frac{\partial u}{\partial t} = -\Delta u \quad \rightarrow \quad u_t = \mathcal{H}_t(u_0) = \exp(-t\mathbf{L}) u_0 \quad \rightarrow \quad -\frac{\partial u_t}{\partial t} = \mathbf{L}u_t = \mathbf{L} \exp(-t\mathbf{L}) u_0$$

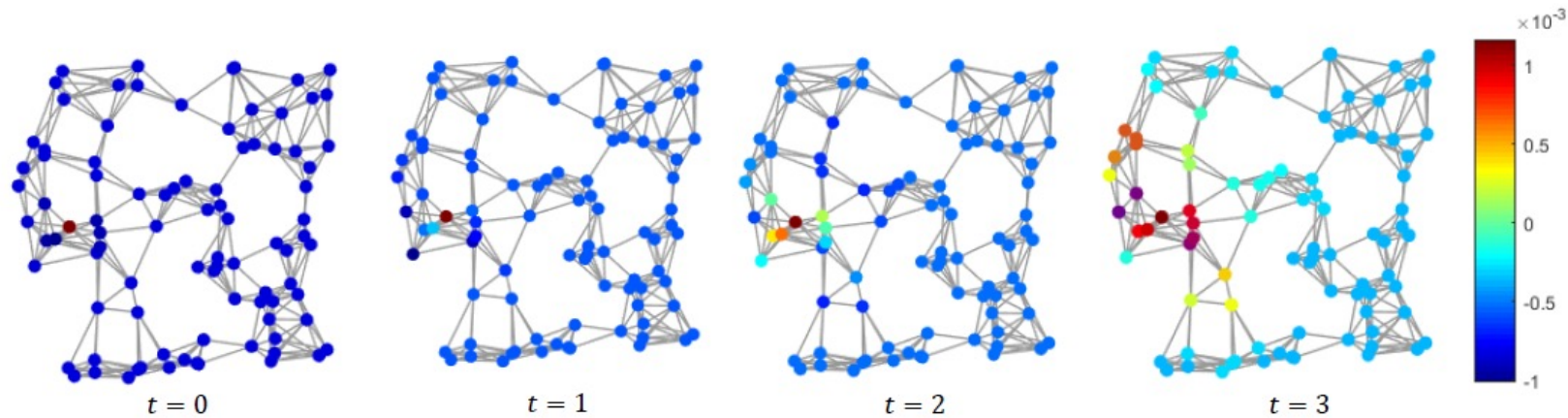
TIDE combines local accuracy with global information propagation by:

$$\mathcal{L}_k^{TIDE}(\mathbf{U}) = \sigma(T_{t_k}(\mathbf{U})\mathbf{W}^{(k)}) = \sigma(\mathbf{L} \exp(-t_k \mathbf{L}) \mathbf{U} \mathbf{W}^k)$$

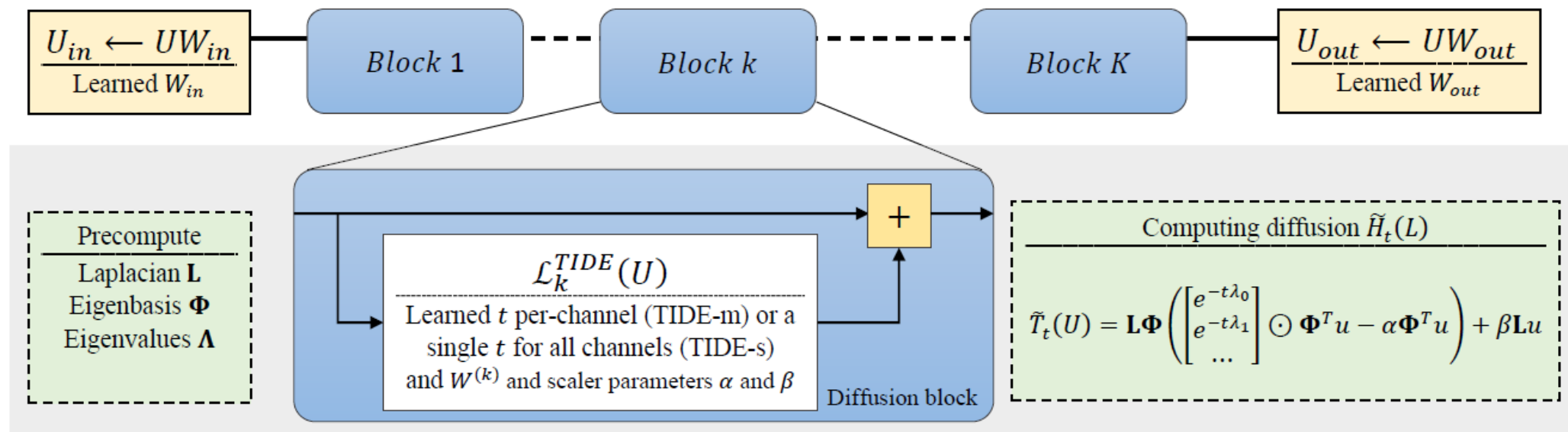
Key idea:

Using *learnable time diffusion* which allows information propagation on the graph

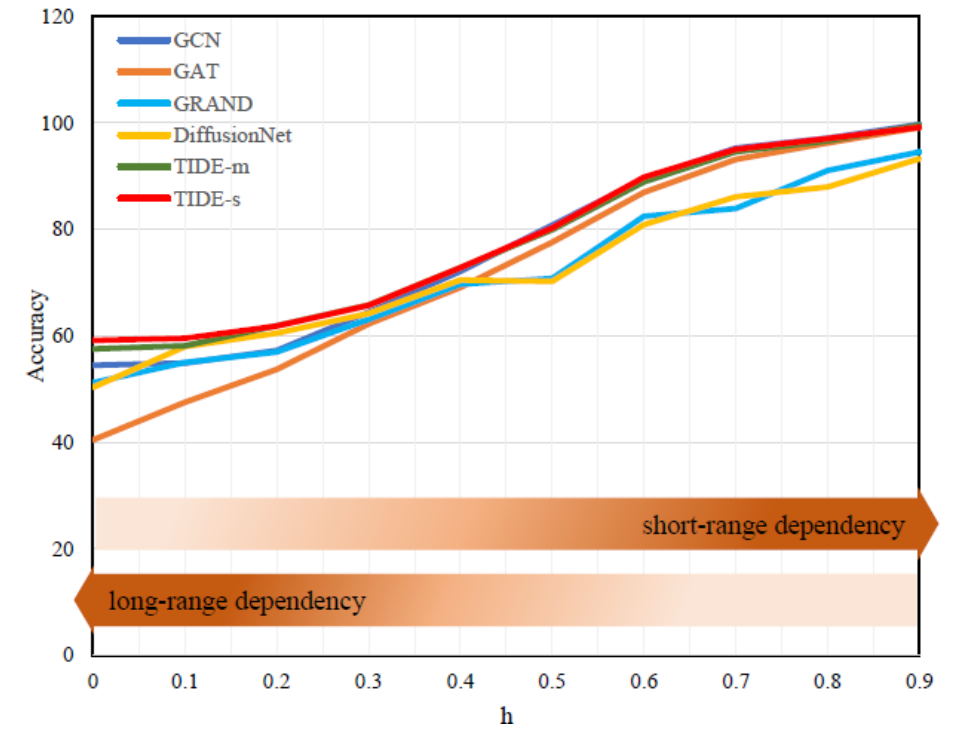
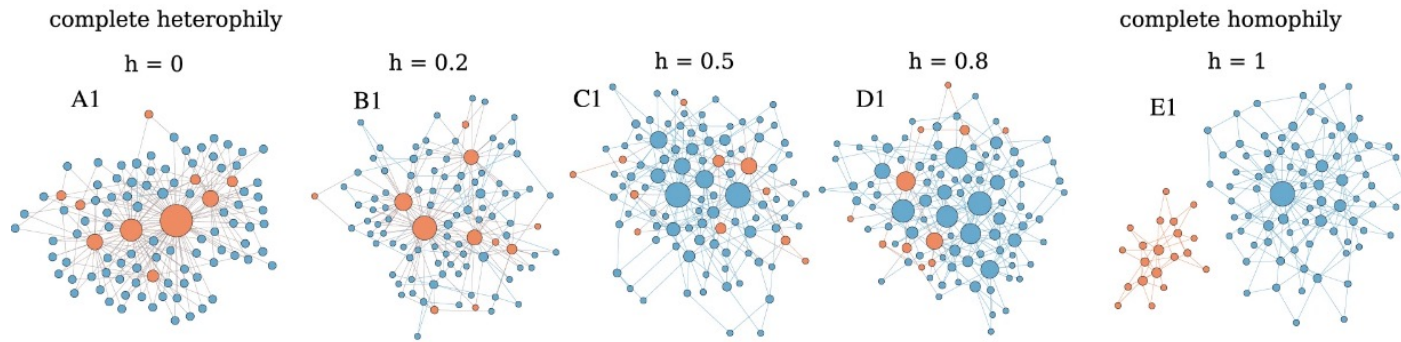
- ↳ variable per-channel spatial support
- ↳ automatically optimized during training



TIDE Architecture



Results: Long Range Communication



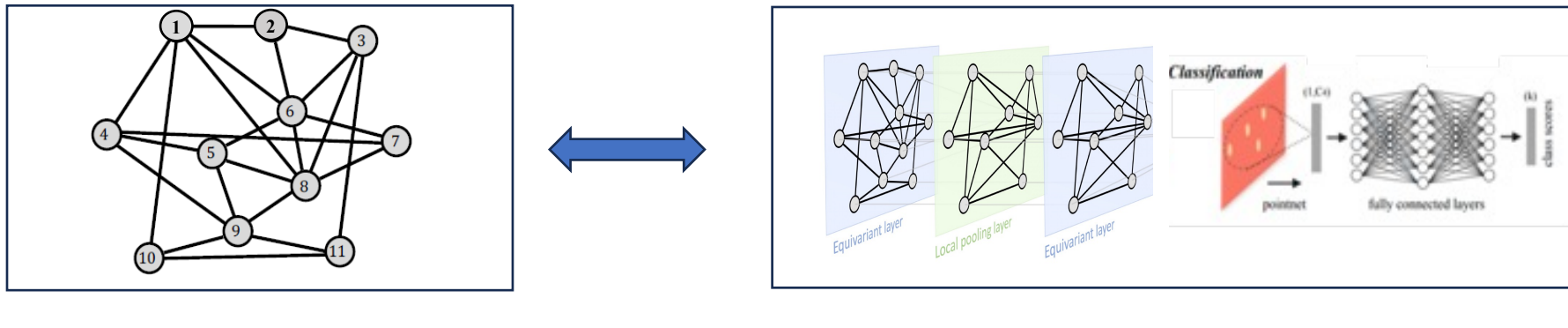
Results:

Node Classification

Model	Cora	Citeseer	Pubmed	CoauthorCS	Computer	Photo	Ogbn-arxiv
GCN (Kipf & Welling, 2017)	83.30±0.36	68.23±0.91	76.78±0.31	90.17±0.50	81.01±0.65	91.71±0.67	65.91±0.12
GAT (Veličković et al., 2017)	81.83±0.42	69.19±0.53	75.49±0.43	90.15±0.35	80.25±0.52	91.57±0.41	54.23±0.22
GRAND (Chamberlain et al., 2021b)	80.71±0.86	68.06±0.18	74.61±0.25	90.59±0.21	72.96±0.49	84.17±0.34	59.29±0.12
GCNII (Chen et al., 2020)	79.94 ± 1.11	<u>70.27±0.32</u>	76.59±0.7	84.27±0.80	32.63±8.6	57.41±3.6	49.87±0.37
ACM (Luan et al., 2022)	81.83±0.12	69.03±0.02	73.3±0.63	91.50±0.13	77±0.65	92.42±0.29	66.23±0.42
DiffusionNet (Sharp et al., 2022)	80.96±0.50	70.00±0.91	73.09±0.15	89.52±0.22	74.72±0.66	87.17±0.26	54.79±0.16
TIDE-m	84.47±0.43	70.32±0.68	77.59±0.04	89.86±0.30	<u>82.11±0.03</u>	91.33±0.47	<u>67.86±1.10</u>
TIDE-s	<u>84.31±0.36</u>	70.24±0.80	<u>77.24±0.62</u>	<u>90.21±0.12</u>	83.01±0.02	<u>92.06±0.51</u>	68.43±0.35

Challenge

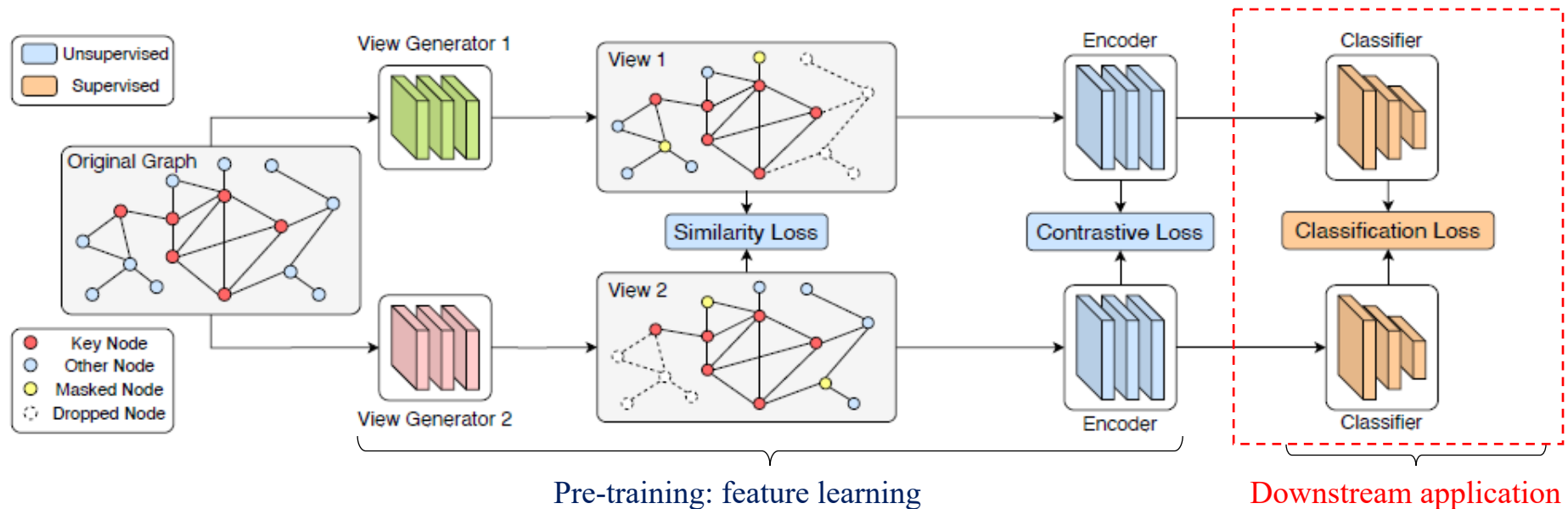
1. GNNs are Limited to *data-rich applications*, not useful in generic tasks!
2. Lack of *generalizable* (transfer) learning on graph



Research direction

Typical representation learning pipeline

Contrastive Learning: powerful feature learning without labels.

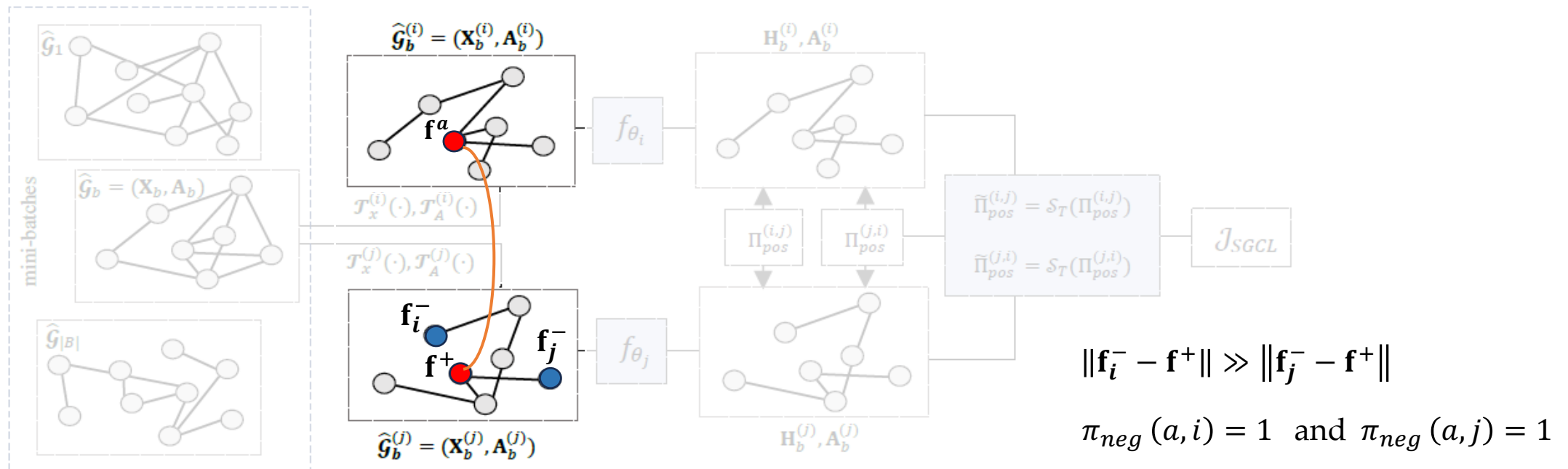


How to learn **informative features** on unlabeled graph (ideally, useful in **downstream applications**) ?

Challenge

Major challenges:

- GCL allocates negative pairs *uniformly*, regardless of their proximity to the true positive.



Our Goals:

- How to integrate proximity information in the contrastive loss?

SGCL: Smoothed Graph Contrastive Learning via Seamless Proximity Integration

Smoothed Graph Contrastive Learning via Seamless Proximity Integration

Maysam Behmanesh¹ Maks Ovsjanikov¹

Abstract

Graph contrastive learning (GCL) aligns node representations by classifying node pairs into positives and negatives using a selection process that typically relies on establishing correspondences within two augmented graphs. The conventional GCL approaches incorporate negative samples uniformly in the contrastive loss, resulting in the equal treatment negative nodes, regardless of their proximity to the true positive. In this paper, we present a Smoothed Graph Contrastive Learning model (SGCL), which leverages the geometric structure of augmented graphs to inject proximity information associated with positive/negative pairs in the contrastive loss, thus significantly regularizing the learning process. The proposed SGCL adjusts the penalties associated with node pairs in the contrastive loss by incorporating three distinct smoothing techniques that result in proximity aware positives and negatives. To enhance scalability for large-scale graphs, the proposed framework incorporates a graph batch-generating strategy that partitions the given graphs into multiple subgraphs, facilitating efficient training in separate batches. Through extensive experimentation in the unsupervised setting on various benchmarks, particularly those of large scale, we demonstrate the superiority of our proposed framework against recent baselines.

1. Introduction

Graph Neural Networks (GNNs) (Gilmer et al., 2017; Kipf & Welling, 2017; Xu et al., 2019b) have developed rapidly by providing the powerful frameworks for the analysis of graph-structured data. A significant portion of GNNs primarily focus on (semi-)supervised learning, which requires

labeling graphs is challenging because they often represent specialized concepts within domains like biology.

Graph Contrastive Learning (GCL), as a new paradigm of Self-Supervised Learning (SSL) (Liu et al., 2023) in the graph domain, has emerged to address the challenge of learning meaningful representations from graph-structured data (Wu et al., 2023; Xie et al., 2023). They leverage the principles of self-supervised learning and contrastive loss (Li et al., 2019) to form a simplified representation of graph-structured data without relying on supervised data.

In a typical GCL approach, several graph views are generated through stochastic augmentations of the input graph. Subsequently, representations are learned by comparing congruent representations of each node, as an anchor instance, with its positive/negative samples from other views (Veličković et al., 2019; Zhu et al., 2020; Hassani & Khasahmadi, 2020). More specifically, the GCL approach initially captures the inherent semantics of the graph to identify the positive and negative nodes. Then, the contrastive loss efficiently pulls the representation of the positive nodes or subgraphs closer together in the embedding space while simultaneously pushing negative ones apart.

Conventional GCL methods follow a straightforward principle when distinguishing between positive and negative pairs: pairs of corresponding points in augmented views are considered positive pairs (similar), while all other pairs are regarded as negative pairs (dissimilar) (Zhu et al., 2020). This strategy ensures that for each anchor node in one augmented view, there exists one positive pair, while all remaining nodes in the second augmented view are paired as negatives.

In contrast to the positive pairs, which are reliably associated with nodes having a similar semantic, there is a significant number of negative pairs that have the potential for false negatives. With this strategy, GCL approaches allocate negative pairs between views uniformly, while we intuitively



Maysam Behmanesh



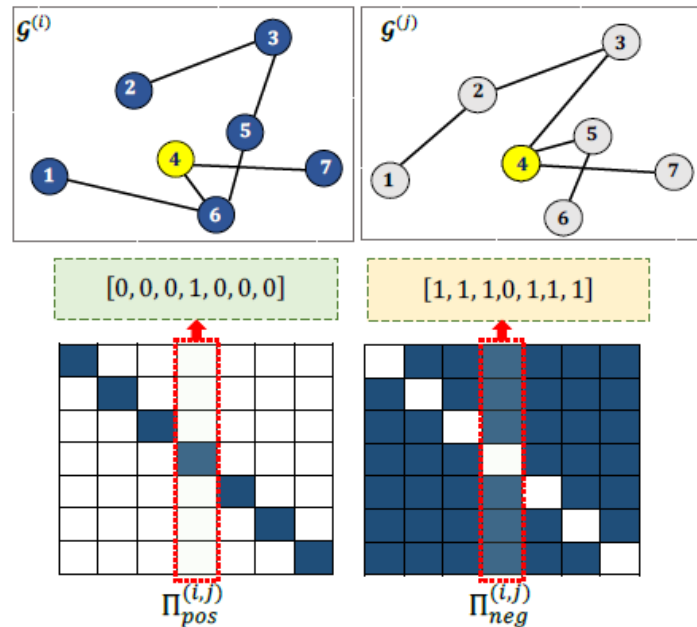
Maks Ovsjanikov

<https://github.com/maysambehtmanesh/SGCL>

“Graphs Smoothed Graph Contrastive Learning via Seamless Proximity Integration,” M. Behmanesh, and M. Ovsjanikov, arxiv, 2024

Our Intuition:

Going beyond simple binary categorization of positive and negative points

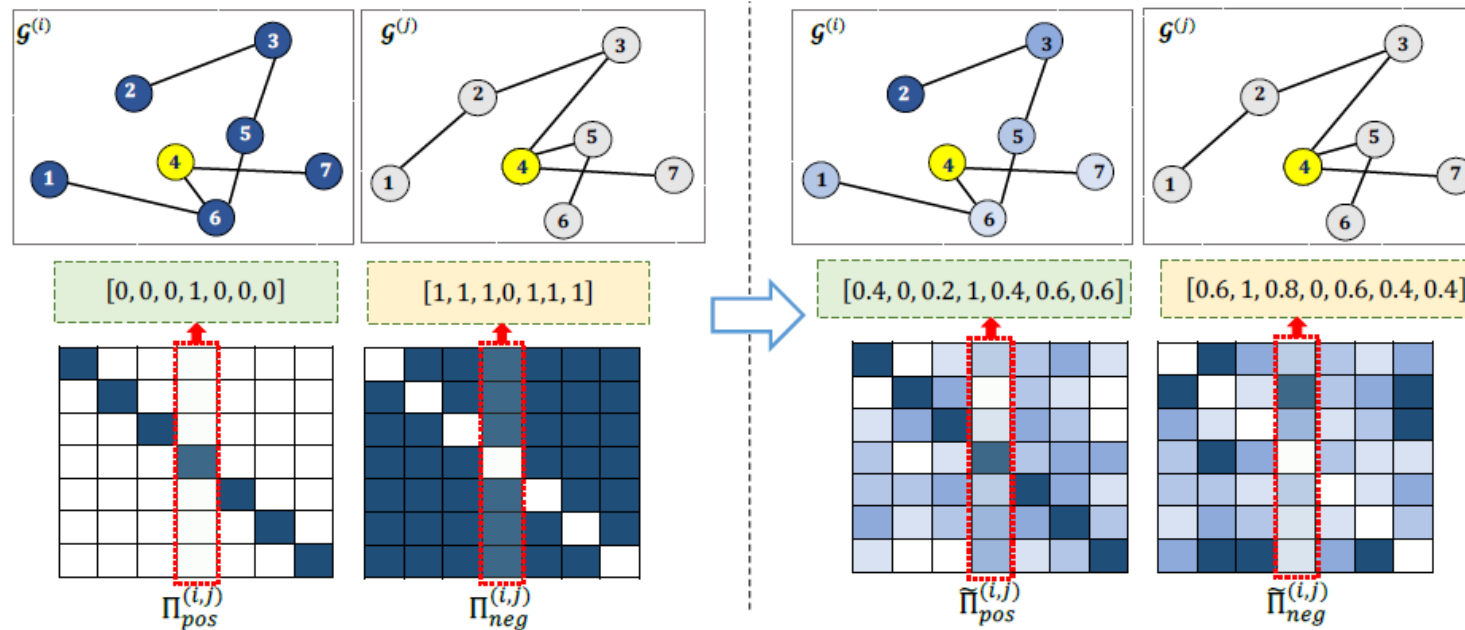


$$\Pi_{pos} \in \{0,1\}^{N \times N}$$

$$\Pi_{neg} \in \{0,1\}^{N \times N}$$

Our Intuition:

Going beyond simple binary categorization of positive and negative points

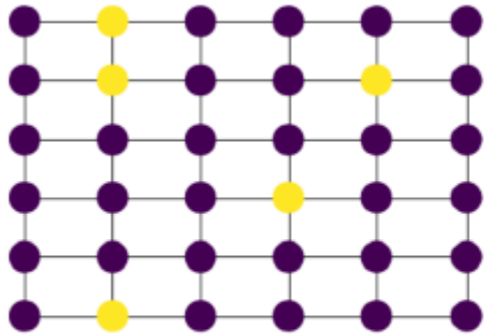


✗ $\Pi_{pos} \in \{0,1\}^{N \times N}$
 $\Pi_{neg} \in \{0,1\}^{N \times N}$

✓ $\tilde{\Pi}_{pos} \in [0,1]^{N \times N}$
 $\tilde{\Pi}_{neg} \in [0,1]^{N \times N}$

Question:

How can proximity information be effectively incorporated into contrastive loss?



Graph $\mathcal{G} = (\mathbf{V}, \mathbf{A})$

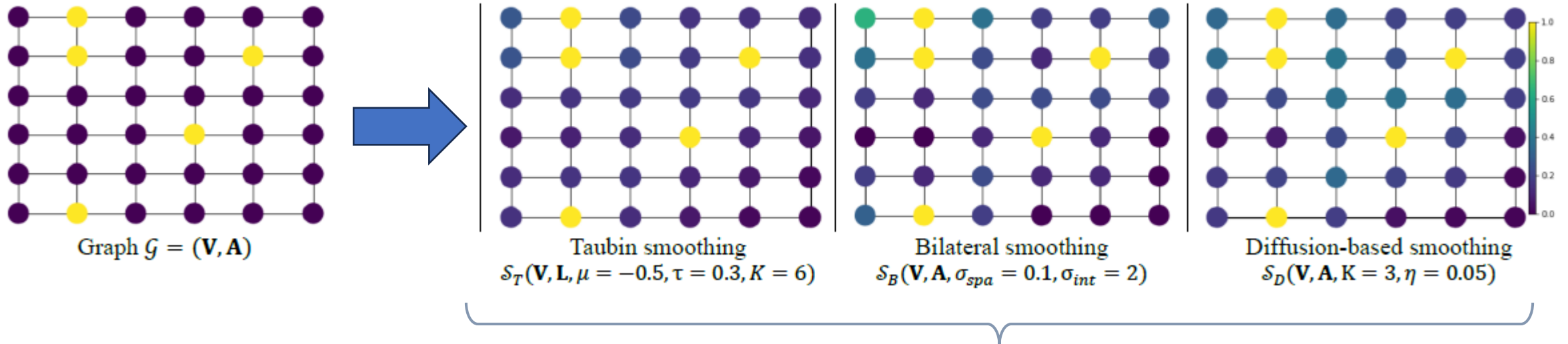
Input:

binary matrix $\Pi \in \{0,1\}^{N \times N}$

Our Intuition:

Applying a smoothing approach for graph

Smoothing involves iteratively updating node values based on the values of their neighboring nodes



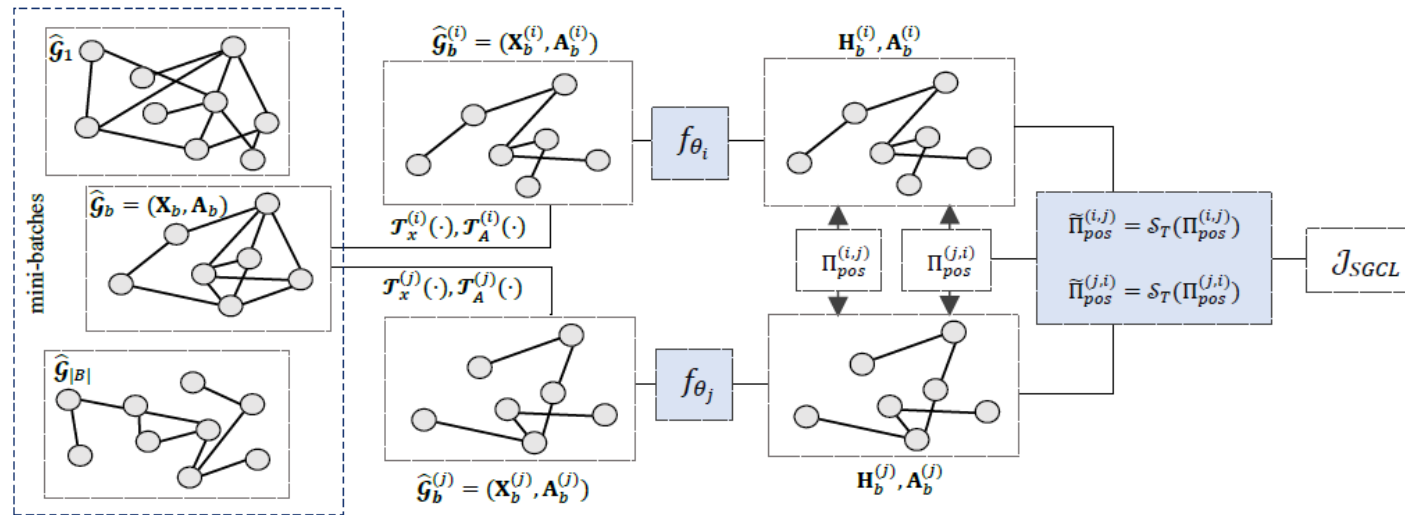
Input:

binary matrix $\Pi \in \{0,1\}^{N \times N}$

Output:

smooth matrix $\tilde{\Pi} \in [0,1]^{N \times N}$

SGCL – Architecture



Contrastive Loss:

$$\mathcal{L}_{SGCL}^{(i,j)} = \|\tilde{\Pi}_{pos}^{(i,j)} \odot (\mathbf{1} - \mathbf{C}^{(i,j)})\|_F^2 + \lambda \|\tilde{\Pi}_{pos}^{(i,j)} \odot \mathbf{C}^{(i,j)}\|_F^2$$

$\mathbf{C}^{(i,j)}$: normalized cosine similarity between the embeddings

Results: Node Classification

Model	Cora	Citeseer	Pubmed	CoauthorCS	Computers	Photo
DGI (Veličković et al., 2019)	76.28±0.04	69.33±0.14	83.79±0.08	91.63±0.08	71.96±0.06	75.27±0.02
GRACE (Zhu et al., 2020)	81.80±0.19	71.35±0.07	85.86±0.05	91.57±0.14	84.77±0.06	89.50±0.06
MVGRL (Hassani & Khasahmadi, 2020)	84.98±0.11	71.29±0.04	85.22±0.04	91.65±0.02	88.55±0.02	91.90±0.08
BGRL (Thakoor et al., 2022)	80.21±1.14	66.33±2.10	81.78±1.06	90.19±0.82	84.24±1.32	89.56±1.01
GBT (Bielak et al., 2022)	79.32±0.31	65.78±1.33	86.35±0.48	91.87±0.07	90.43±0.18	92.23±0.18
CGRA (Duan et al., 2023)	82.71±0.01	69.23±1.19	82.15±0.46	91.26±0.27	89.76±0.36	91.54±1.06
GRLC (Peng et al., 2023)	83.50±0.24	70.02±0.16	81.20±0.20	90.36±0.27	88.54±0.23	91.80±0.77
SGCL-T	84.45±0.04	71.26±0.06	84.11±0.08	92.14±0.09	86.81±0.01	92.71±0.05
SGCL-B	85.08±0.12	72.77±0.33	83.67±0.06	92.16±0.15	88.24±0.05	92.43±0.03
SGCL-D	84.47±0.25	70.32±0.04	85.22±0.02	92.04±0.05	84.98±0.34	90.09±0.11

Model	ogbn-arxiv	ogbn-products	ogbn-proteins
DGI (Veličković et al., 2019)	67.07±0.5	68.68±0.6	94.11±0.1
GRACE (Zhu et al., 2020)	67.92±0.4	72.10±0.7	94.11±0.2
MVGRL (Hassani & Khasahmadi, 2020)	60.68±0.5	69.90±0.9	93.87±0.3
BGRL (Thakoor et al., 2022)	63.88±0.2	66.23±0.5	92.94±0.3
GBT (Bielak et al., 2022)	69.05±0.3	65.74±0.4	94.07±0.3
SGCL-T	69.30±0.5	75.97±0.1	94.64±0.2
SGCL-B	69.24±0.3	74.33±0.4	93.55±0.2
SGCL-D	69.03±0.4	74.15±0.2	93.19±0.1

Results: Graph Classification

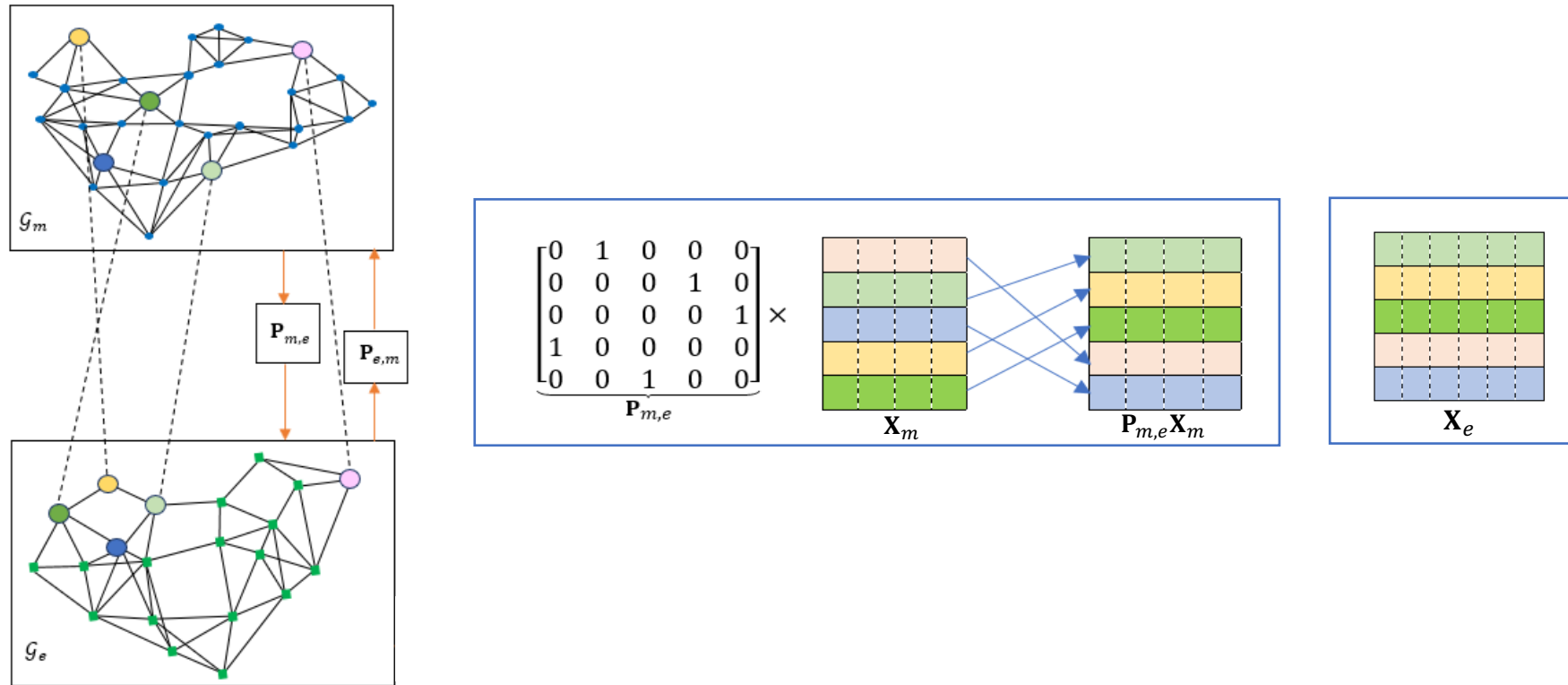
Model	IMDB-Binary	PTC	MUTAG	PROTEINS	ENZYMES
InfoGraph (Sun et al., 2019)	73.0±0.9	61.7±1.4	89.0±1.1	74.4±0.3	50.2±1.4
GraphCL (You et al., 2020)	71.1±0.4	63.6±1.8	86.8±1.3	74.4±0.5	55.1±1.6
MVGRL (Hassani & Khasahmadi, 2020)	74.2±0.7	62.5±1.7	89.7±1.1	71.5±0.3	48.3±1.2
AD-GCL (Suresh et al., 2021)	71.5±1.0	61.2±1.4	86.8±1.3	75.0±0.5	42.6±1.1
BGRL (Thakoor et al., 2022)	72.8±0.5	57.4±0.9	86.0±1.8	77.4±2.4	50.7±9.0
LaGraph (Xie et al., 2022)	73.7±0.9	60.8±1.1	90.2±1.1	75.2±0.4	40.9±1.7
CGRA (Duan et al., 2023)	75.6±0.5	65.7±1.8	91.1±2.5	76.2±0.6	61.1±0.9
SGCL-T	75.2±2.8	64.0±1.6	89.0±2.3	79.4±1.9	65.3±3.6
SGCL-B	73.2±3.7	62.5±1.8	87.0±2.8	81.6±2.3	63.7±1.6
SGCL-D	75.8±1.9	62.6±1.4	86.0±2.6	81.5±2.3	64.3±2.2

Ongoing Projects...

Large Scaled Graph Matching

Challenges

Complexity: Optimal Transport (OT)-based loss, (like GWD) can be computationally expensive
Scalability: OT-based methods may not scale well to large-scale graph datasets



- ✓ Large scaled graph matching with learned features
- ✓ Transferable functional maps for graph matching

Future Projects...

- 1- Large-scale multimodal data
- 2- Multimodal time-varying data
- 3- Applications in practical domains (multimodal sentiment analysis, multimedia retrieval, visual question answering ...)

Thank You For Your Attention

Questions?



Acknowledgements:

TIDE and SGCL work are supported by the ERC Starting Grant No. 758800 (EXPROTEA), the ANR AI Chair AIGRETTE.

M-GWCN is supported by MIAI@Grenoble Alpes under Grant ANR-19-P3IA-0003.