Self Introduction

Jingwei Guo

Email: jingweiguo19@outlook.com

Homepage: jingweio.github.io

Basic Information

Alibaba Group

Aug. 2024 - Present

- Senior Algorithm Engineer in Taobao & Tmall Group
- Responsibilities: Advertisement Reranking System

University of Liverpool

Dec. 2019 - Jul. 2024

- Ph.D. in CS & EE
- Advisors: Prof. Kaizhu Huang, Prof. Xinping Yi

Kaizhu Huang

Professor of Electrical and Computer Engineering at Duke Kunshan University

DKU Faculty

kaizhu.huang@dukekunshan.edu.cn

Dr. Xinping Yi

Professor

National Mobile Communications Research Laboratory

Southeast University

Nanjing, China

Email: xyi at seu dot edu dot cn

Research Projects

1. Graph Learning Against Homophilous Assumptions

Jingwei Guo, et.al. Rethinking Spectral Graph Neural Networks with Spatially Adaptive Filtering. *TNNLS (Under 2nd Review) 2025.*

Jingwei Guo, et.al. ES-GNN: Generalizing Graph Neural Networks Beyond Homophily with Edge Splitting. *TPAMI 2024.*

Jingwei Guo, et.al. Graph Neural Networks with Diverse Spectral Filtering. *WWW 2023*.

Jingwei Guo, et.al. Learning Disentangled Graph Convolutional Networks Locally and Globally. *TNNLS 2022.*

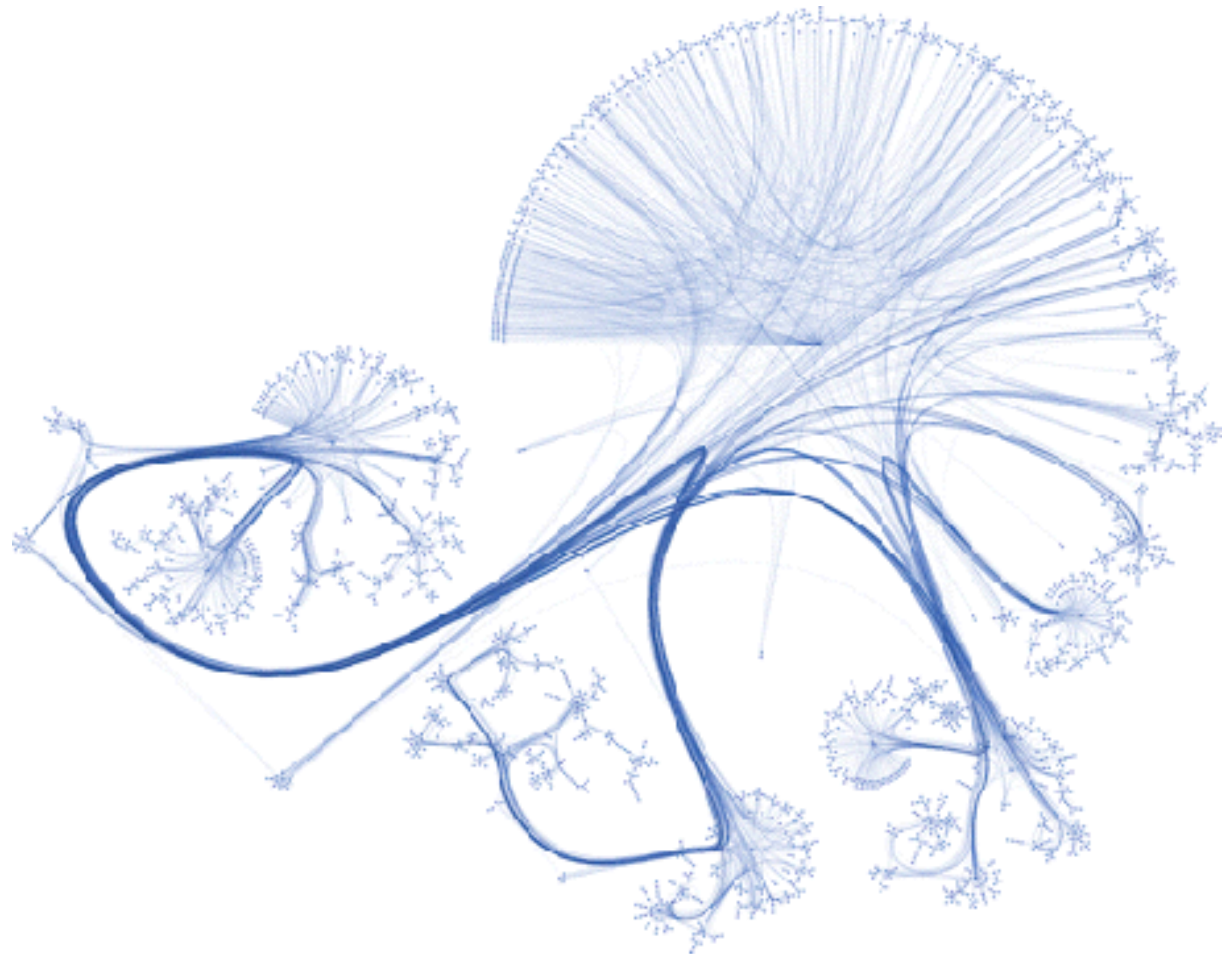
2. Transfer Learning Under Distribution Shifts

Zixian Su*, **Jingwei Guo***, et.al. Un-mixing Test-time Adaptation under Heterogeneous Data Streams. *TKDE (Under Review) 2025.*

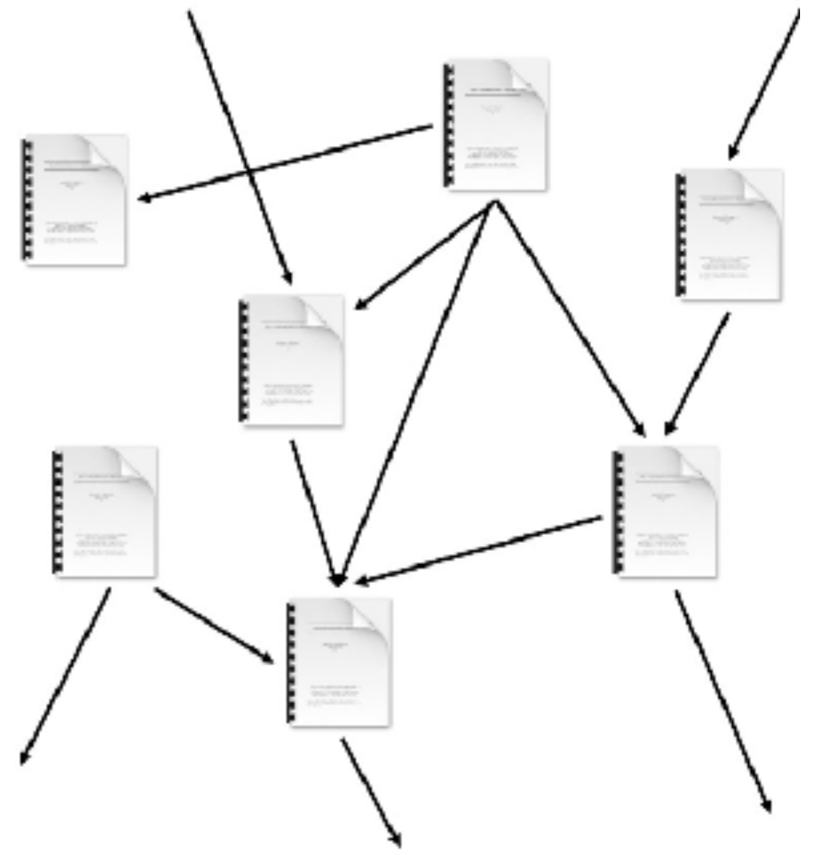
Zixian Su*, **Jingwei Guo***, et.al. Navigating Distribution Shifts in Medical Image Analysis. *ACM Comput. Surv (Under Review) 2025.*

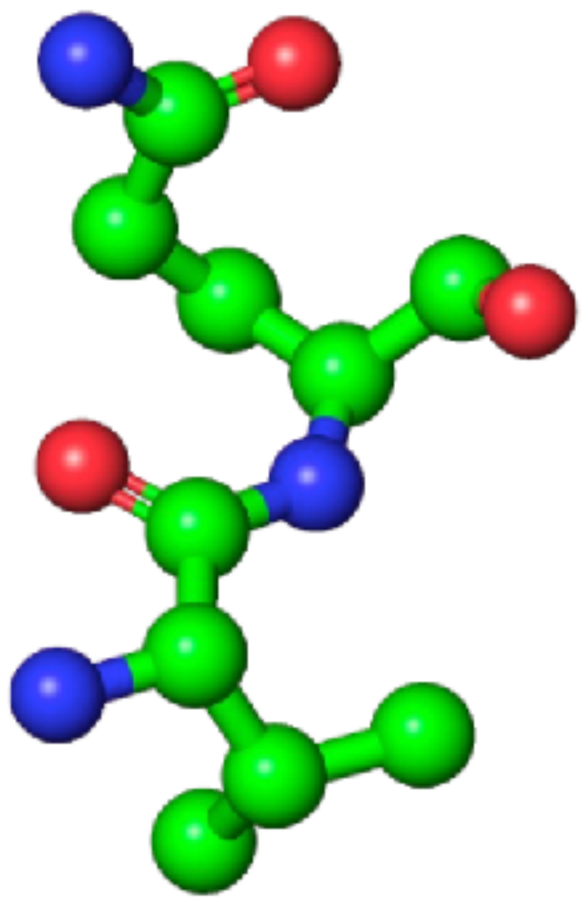
Zixian Su, **Jingwei Guo**, et.al. Unraveling Batch Normalization for Realistic Test-Time Adaptation. *AAAI 2024.*

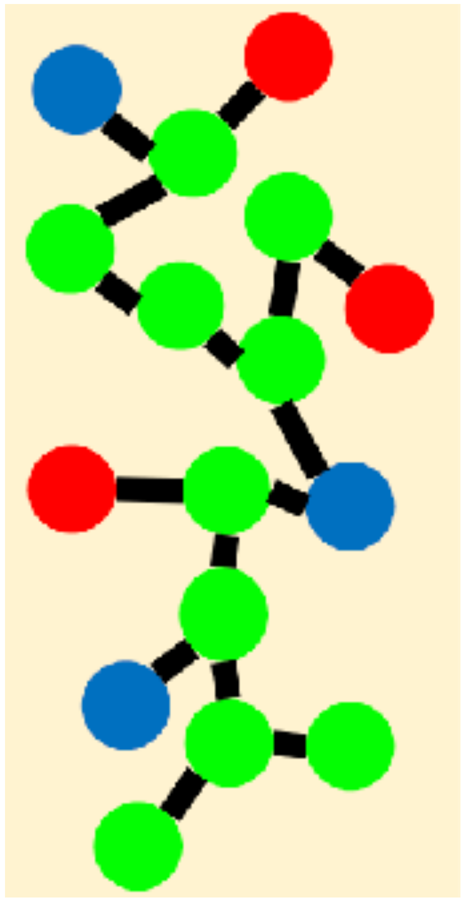
Graph Learning Against Homophilous Assumptions

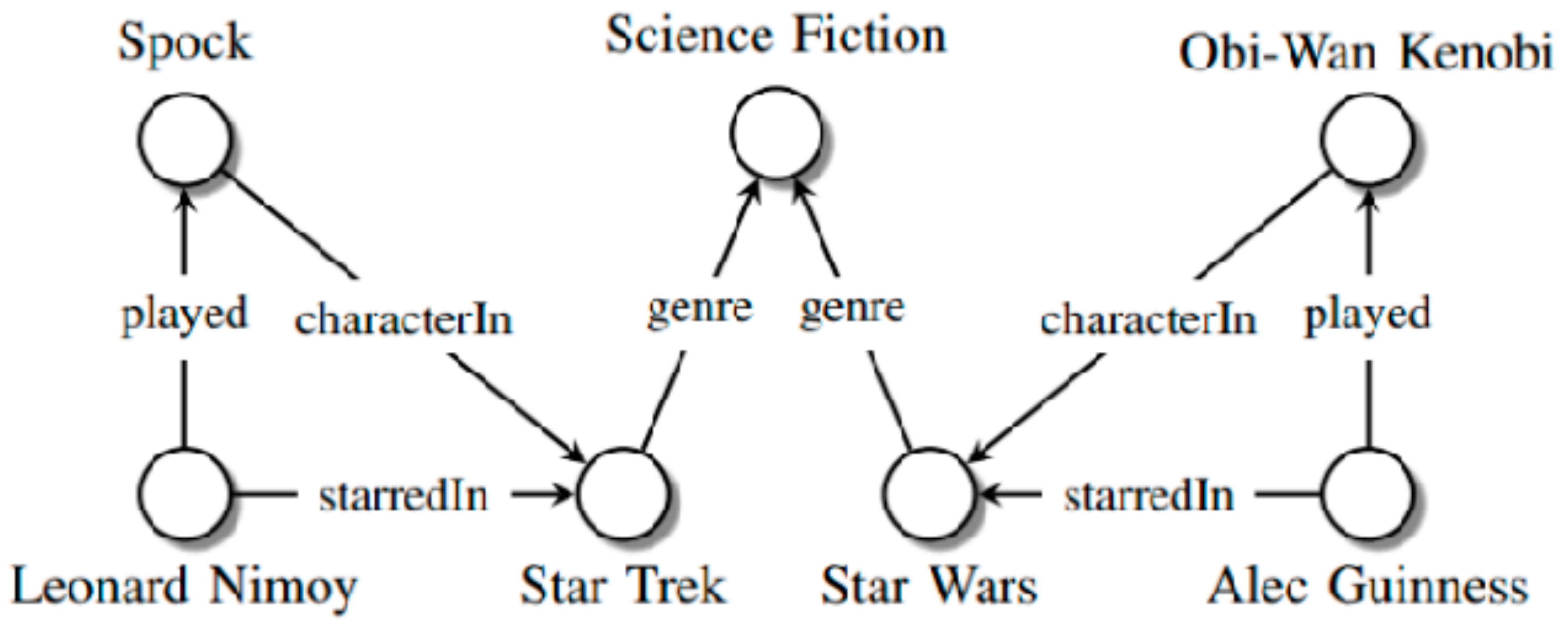


Research Background — Graphs



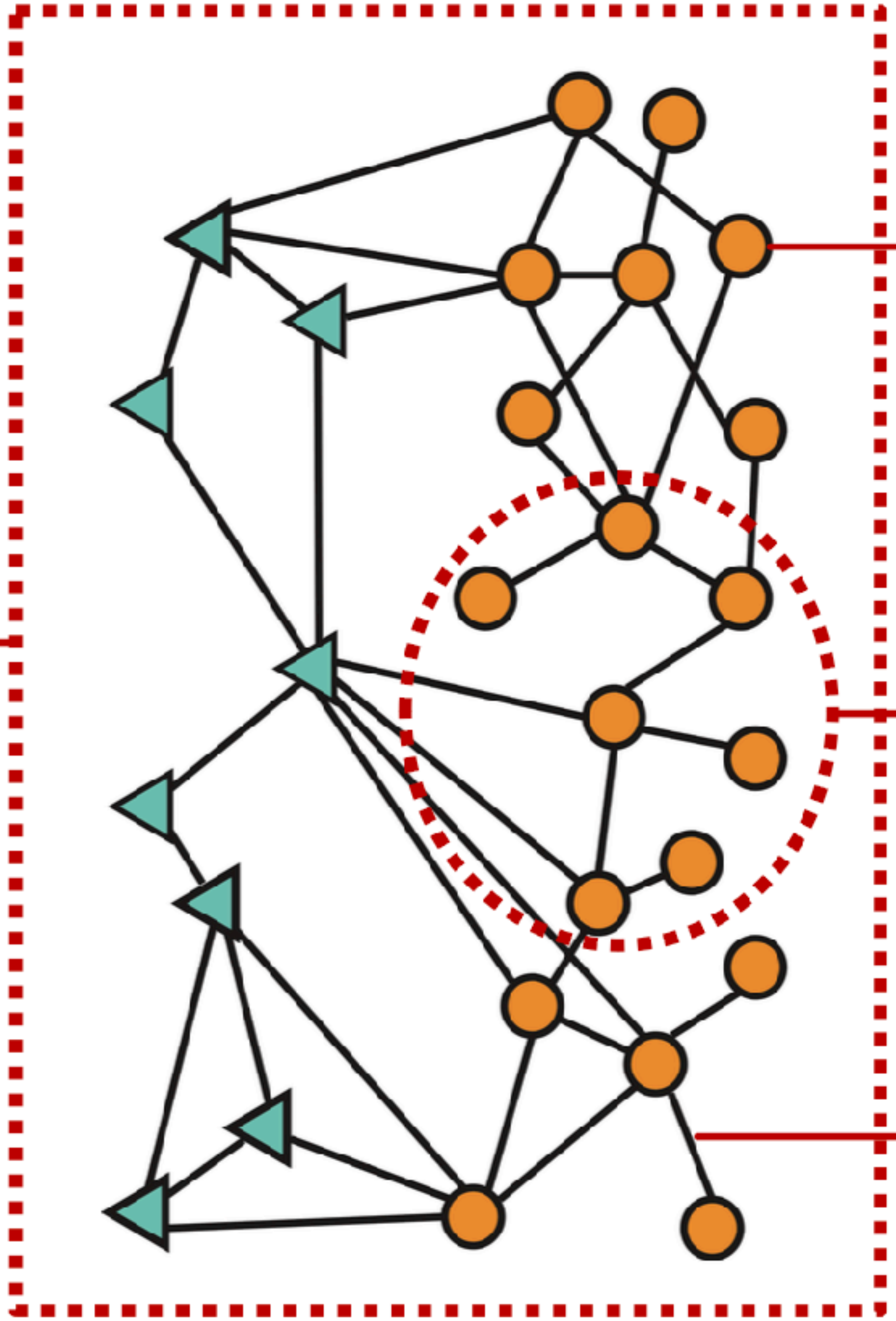






Graphs are a general language describing and analyzing entities with relations or interactions.

Research Background — Tasks on Graphs



Node-level

e.g., Node Classification

Graph-level

e.g., Graph Classification /Generation

Subgraph-level

e.g., Community Detection

Edge-level

e.g., Recommendation

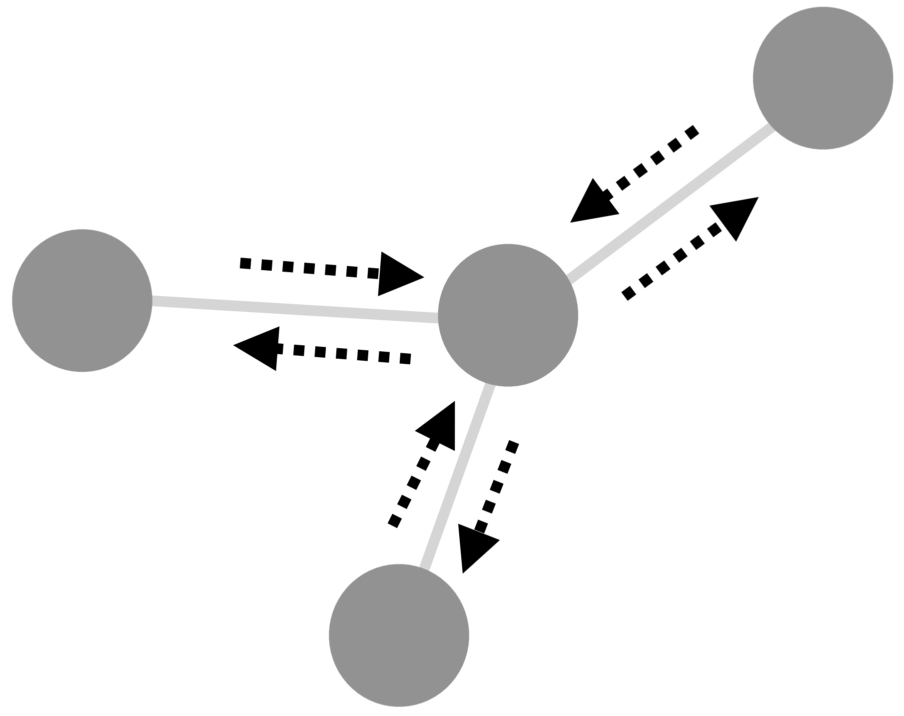
http://cs224w.stanford.edu/

Core Insights

- Integrate both node features and graph topology via either a message passing framework or a graph filtering operation.

  - Spatial-based Methods

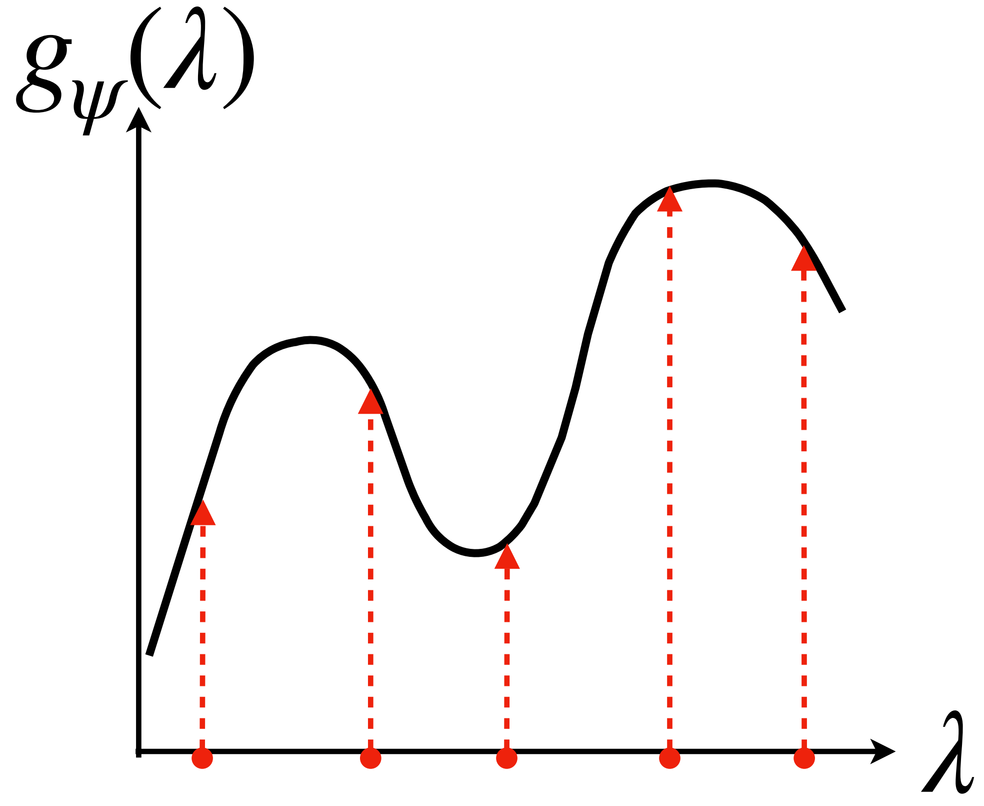
$$\mathbf{z}_i = f_{upd}(\mathbf{x}_i, f_{agg}(\{\mathbf{x}_j \mid \forall v_j \in N_i\}))$$



Message Passing

  - Spectral-based Methods

$$\mathbf{Z} = \mathbf{U} g_{\psi}(\boldsymbol{\Lambda}) \mathbf{U}^T \mathbf{X}$$



Graph Filtering

Optimization Objective

- Can be interpreted as different solutions to the same graph denoising problem:

$$\arg\min_{\mathbf{Z}} \alpha \|\mathbf{X} - \mathbf{Z}\|_2^2 + (1-\alpha)\mathrm{tr}(\mathbf{Z}^T \hat{\mathbf{L}} \mathbf{Z})$$

$$\sum_{(v_i, v_j) \in E} \|deg_i^{-\frac{1}{2}} \mathbf{z}_i - deg_j^{-\frac{1}{2}} \mathbf{z}_j\|_2^2$$

keep close to the original features

smooth node features across the graph

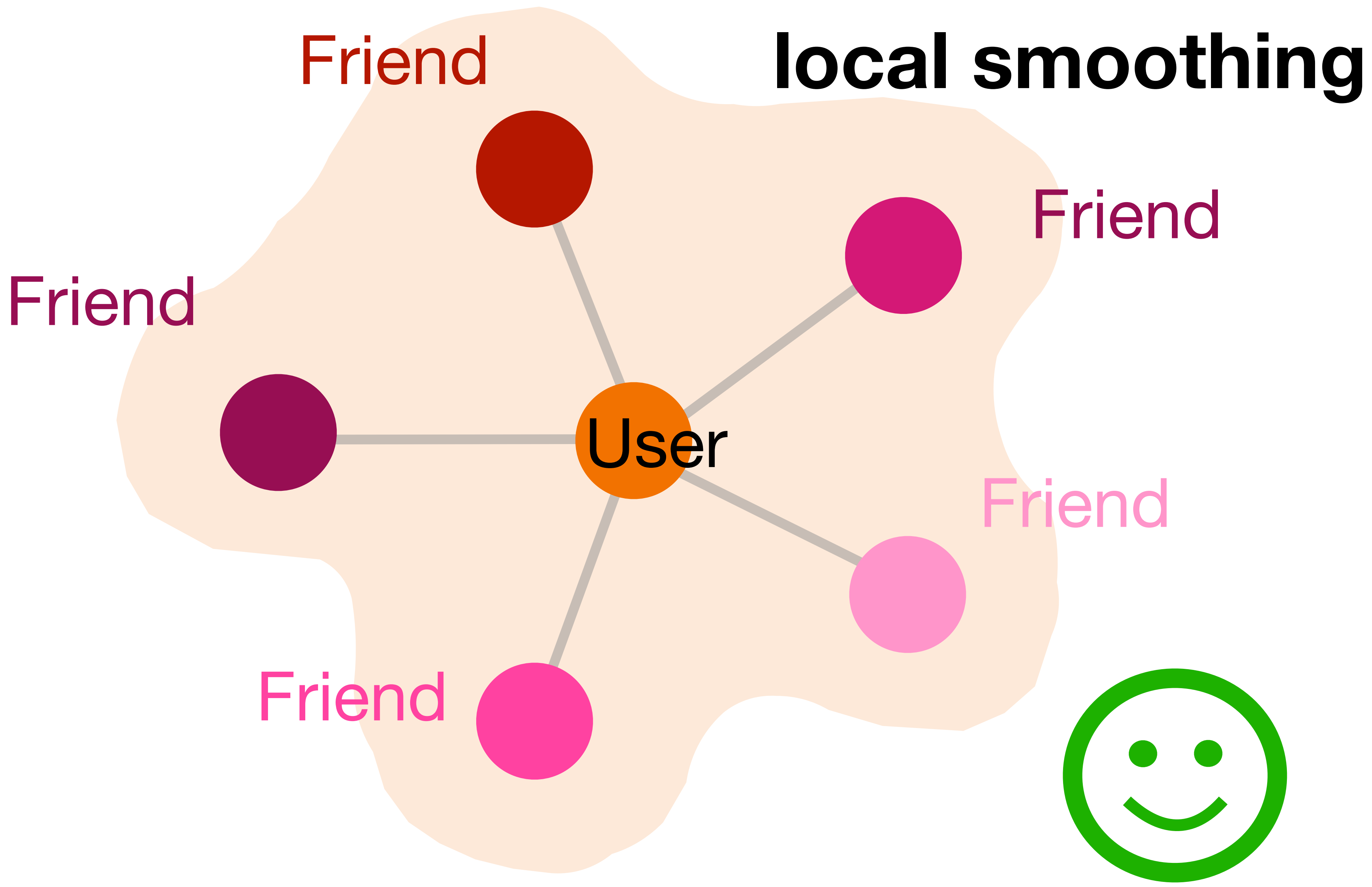
Ma, Y., et.al. A unified view on graph neural networks as graph signal denoising. *CIKM, 2021.* Zhu, M., et.al. Interpreting and unifying graph neural networks with an optimization framework. *WWW, 2021.*

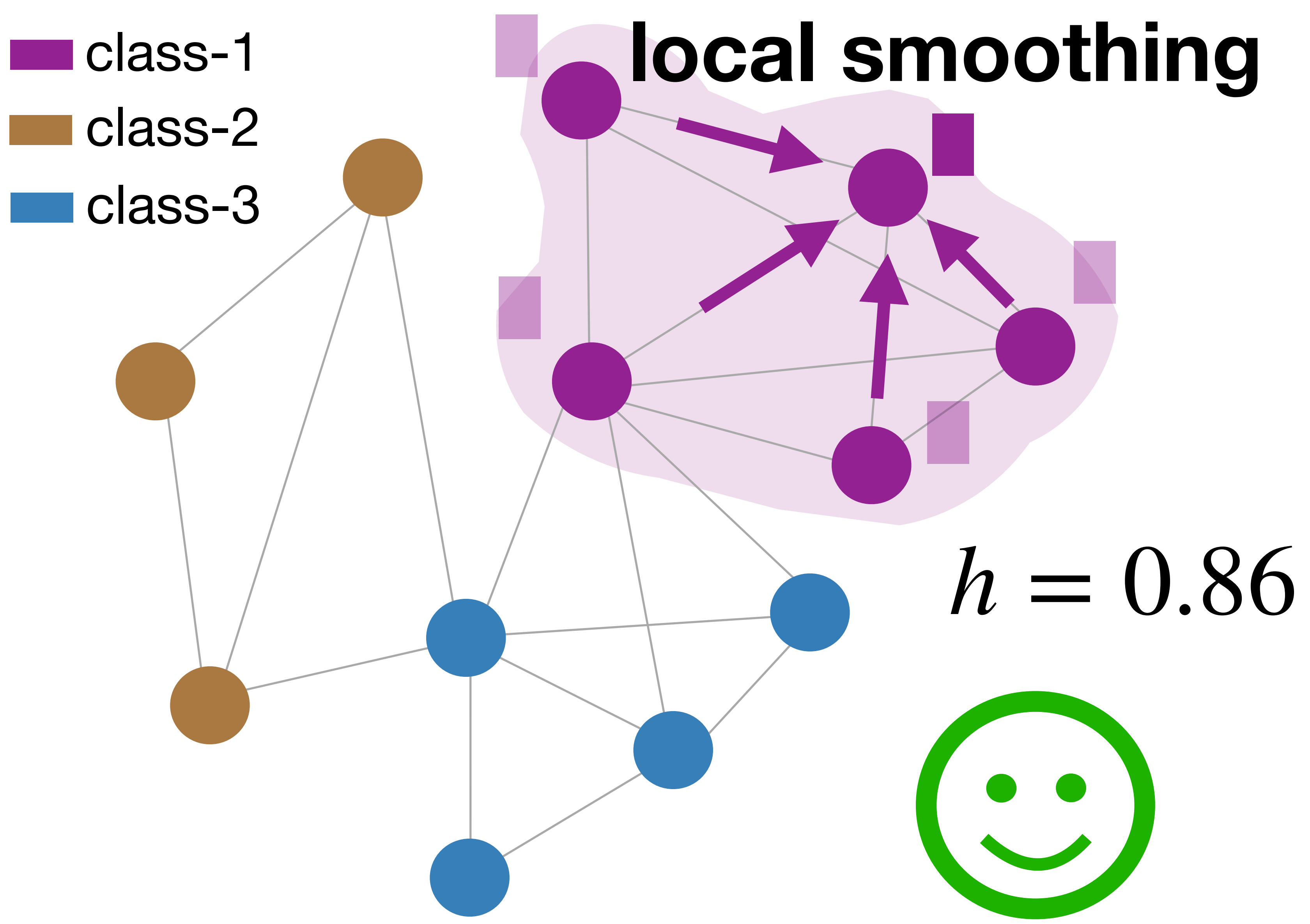
Research Motivation & Challenges

Idealized Scenarios

h = # intra-class edges / # total edges

- Homogenous Node Relationships
- Homophilic Linking Patterns





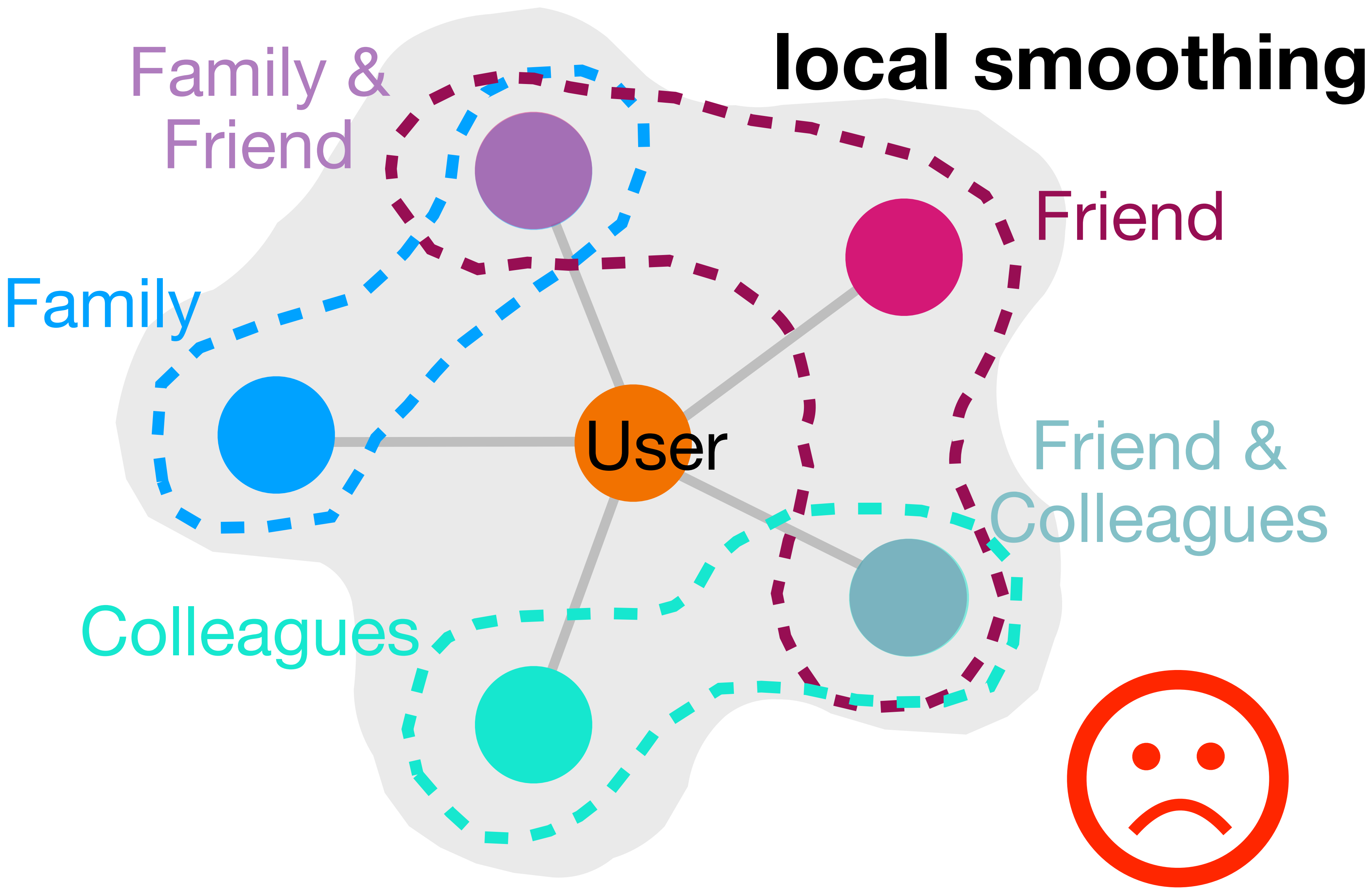
Conventional GNNs assume <u>homogenous node interactions</u> and employ <u>neighborhood smoothing</u>.

Research Motivation & Challenges

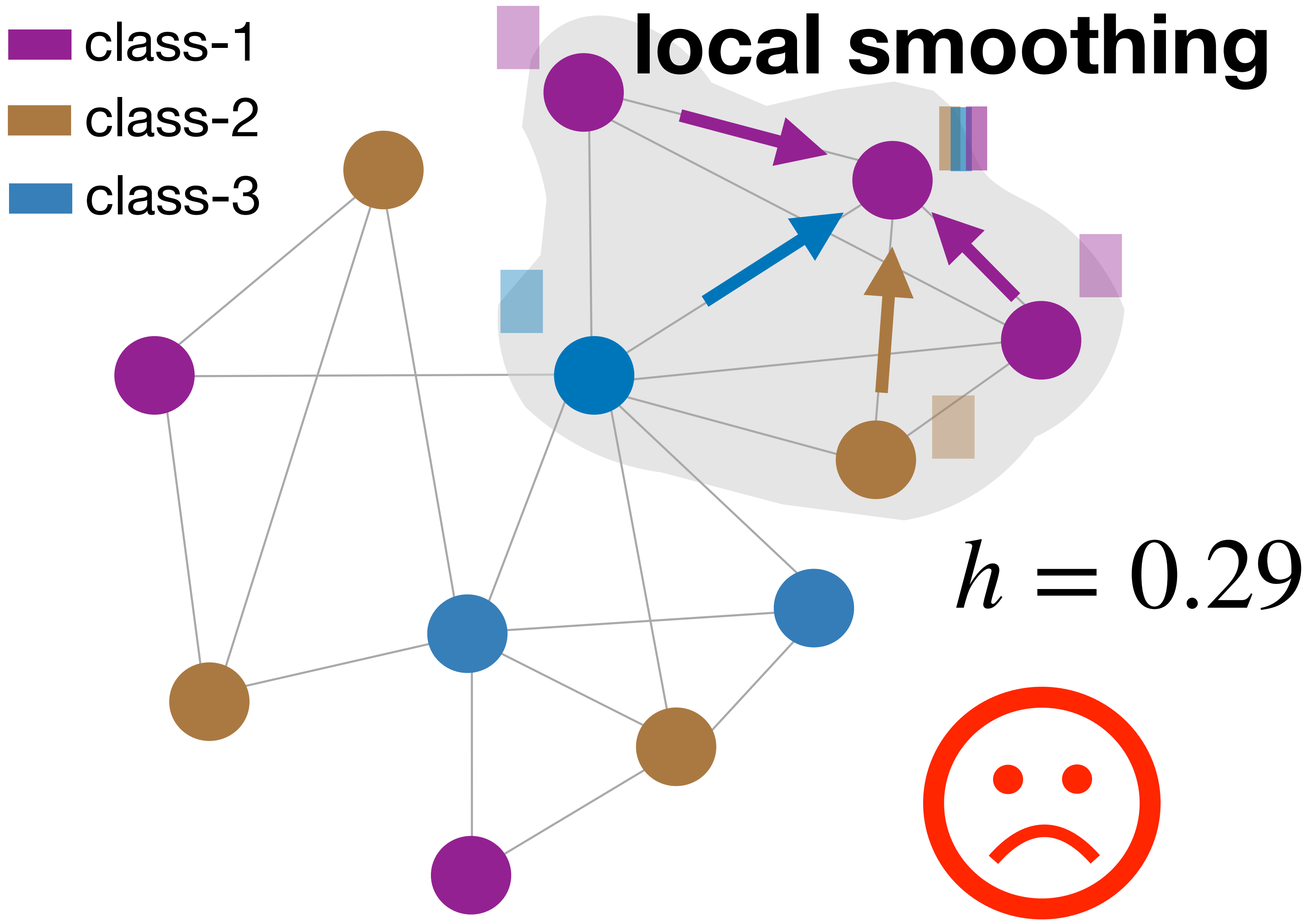
Real-world Scenarios

h = # intra-class edges / # total edges

- Entangled Node Relationships



- Heterophilic Linking Patterns

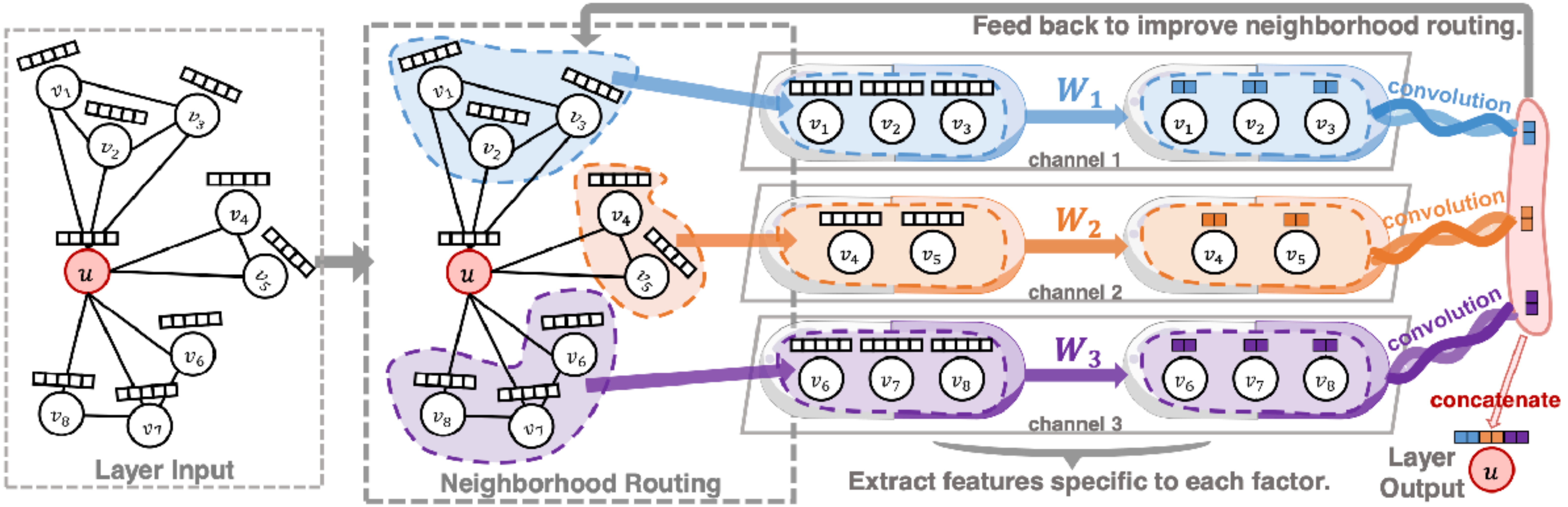


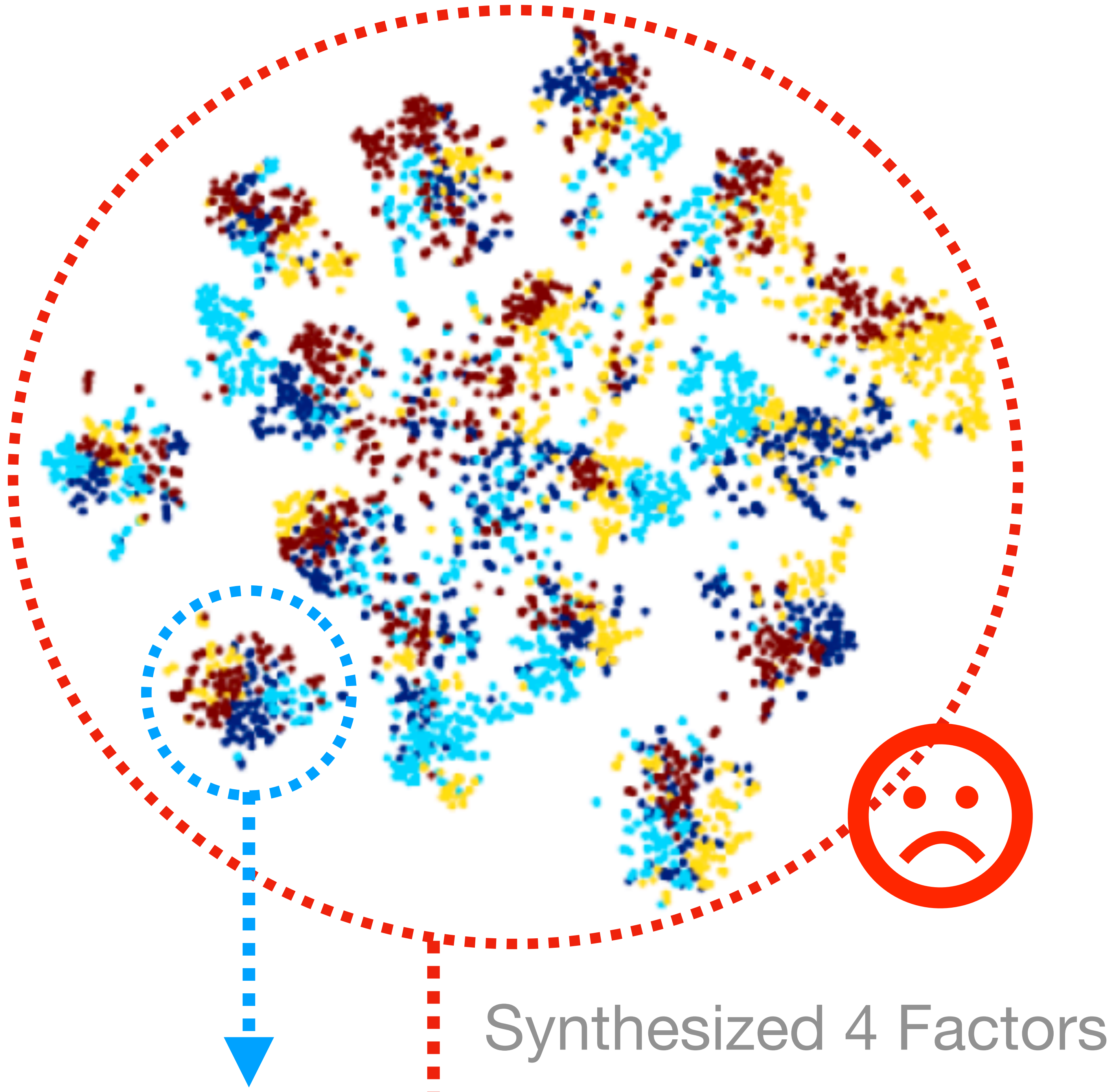
Conventional GNNs assume homogenous node interactions and employ neighborhood smoothing.

Solution #1: Local-Global Graph Disentanglement

Proposed Solution #1: Local-Global Graph Disentanglement

Our Findings (on existing graph disentanglement):





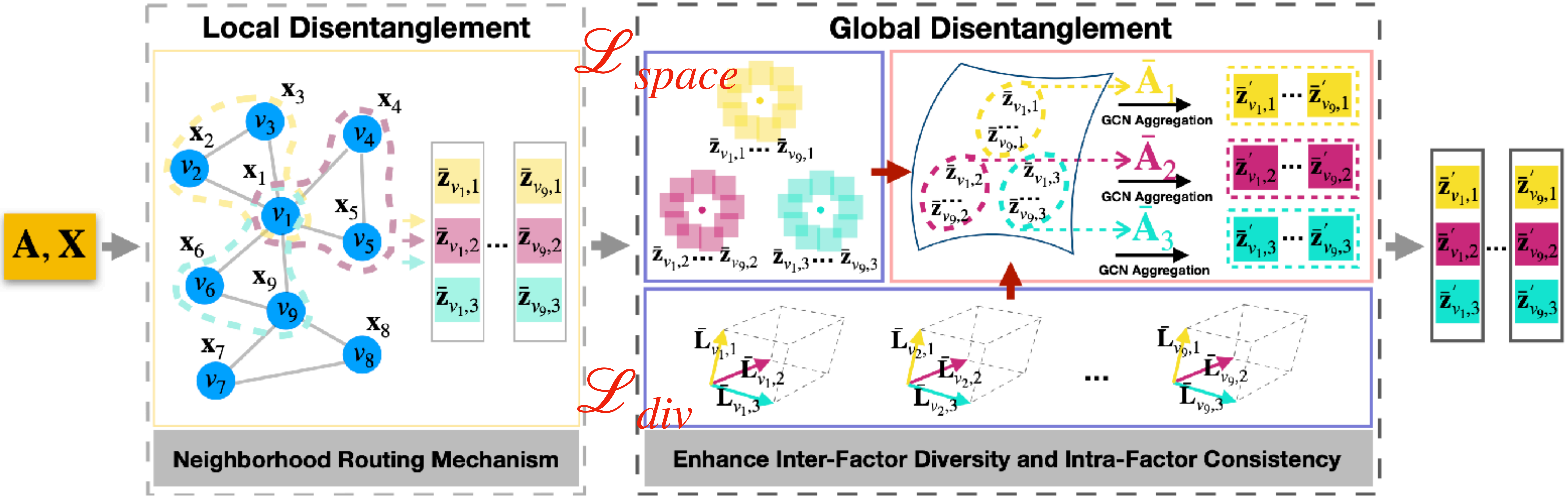
DisenGCN relies on a routing mechanism for local neighborhood partition.

"disentangled locally yet entangled globally"

Jianxin Ma, et.al., Disentangled Graph Convolutional Networks. *ICML, 2021*.

Proposed Solution #1: Local-Global Graph Disentanglement

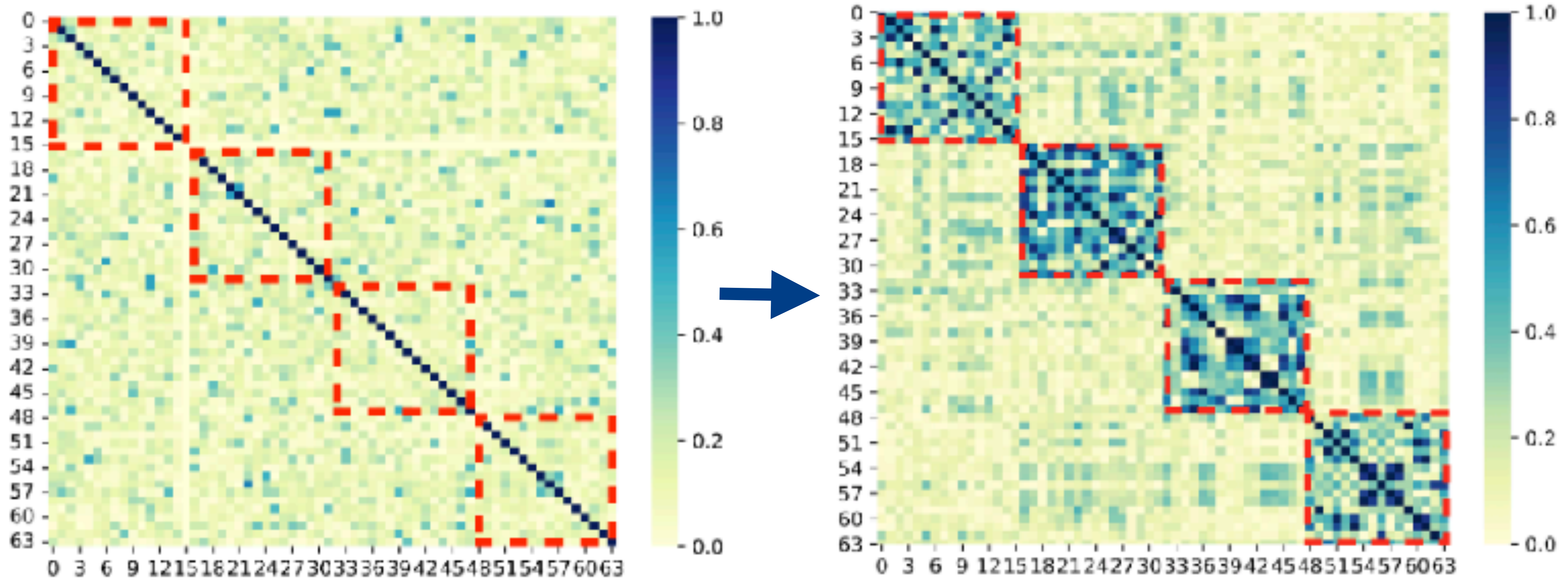
Our Modifications:



- Gaussian Mixture-based Latent Space Regularization
  - Global Message Passing along Latent Node Relations

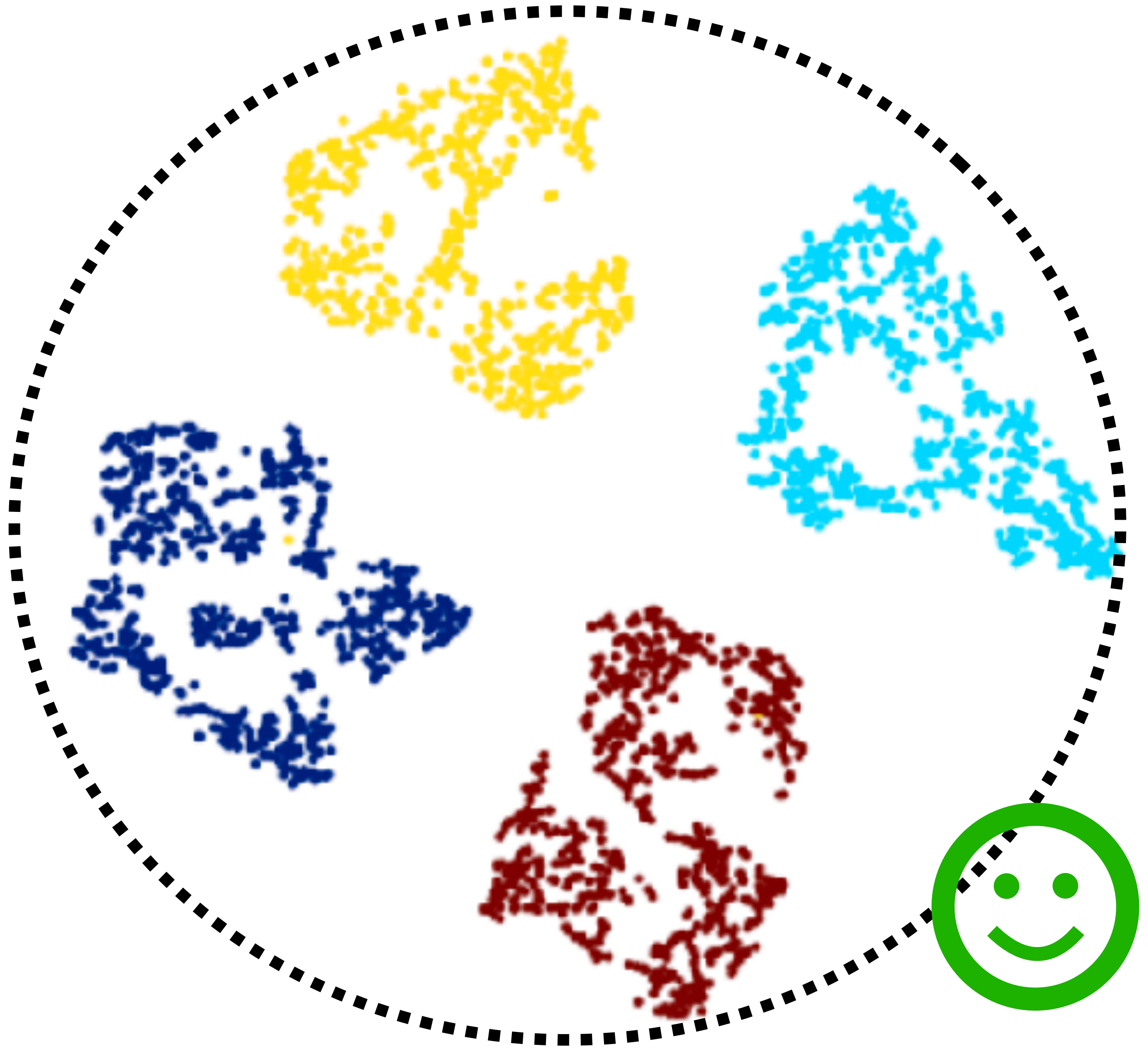
Jingwei Guo, et.al. Learning Disentangled Graph Convolutional Networks Locally and Globally. *TNNLS, 2022*.

Empirical Results



DisenGCN LGD-GCN (Ours)

Notable Block-wise Pattern



Synthesized 4 Factors

"disentangled both locally and globally"

Accuracy +7.4% on Social Nets

Methods	Blogcatalog		Flickr	
MoNet [69]	74.7±0.4	74.6±0.5	61.7±0.7	61.6±1.0
GCN [6]	73.8±0.3	72.9±0.4	56.6±0.4	57.6±0.3
GraphSAGE [10]	73.7±0.3	73.0±0.4	56.3±0.4	57.0±0.4
GAT [11]	56.7±5.0	57.5±3.2	45.1±1.0	45.1±1.4
SGC [12]	74.5±0.3	73.7±0.4	61.4±0.2	60.6±0.3
JK-Net [13]	76.5±0.3	75.8±0.5	64.6±0.4	64.1±0.4
DisenGCN [17]	86.5±1.3	86.4±1.2	75.8±0.6	76.7±0.6
IPGDN [18]	86.9±0.9	86.1±1.1	75.9±0.5	76.8±0.6
FactorGCN [47]	78.4±1.3	77.6±2.1	47.0±1.7	45.4±2.0
LGD-GCN (ours)	**93.7±0.4**	**93.9±0.4**	**85.5±0.6**	**84.0±0.8**

Jingwei Guo, et.al. Learning Disentangled Graph Convolutional Networks Locally and Globally. *TNNLS, 2022*.

Jianxin Ma, et.al.Disentangled Graph Convolutional Networks. *ICML, 2021*.

Solution #2: Edge-Splitting Graph Neural Networks

Jingwei Guo, et.al. ES-GNN: Generalizing Graph Neural Networks Beyond Homophily with Edge Splitting. *TPAMI, 2024*.

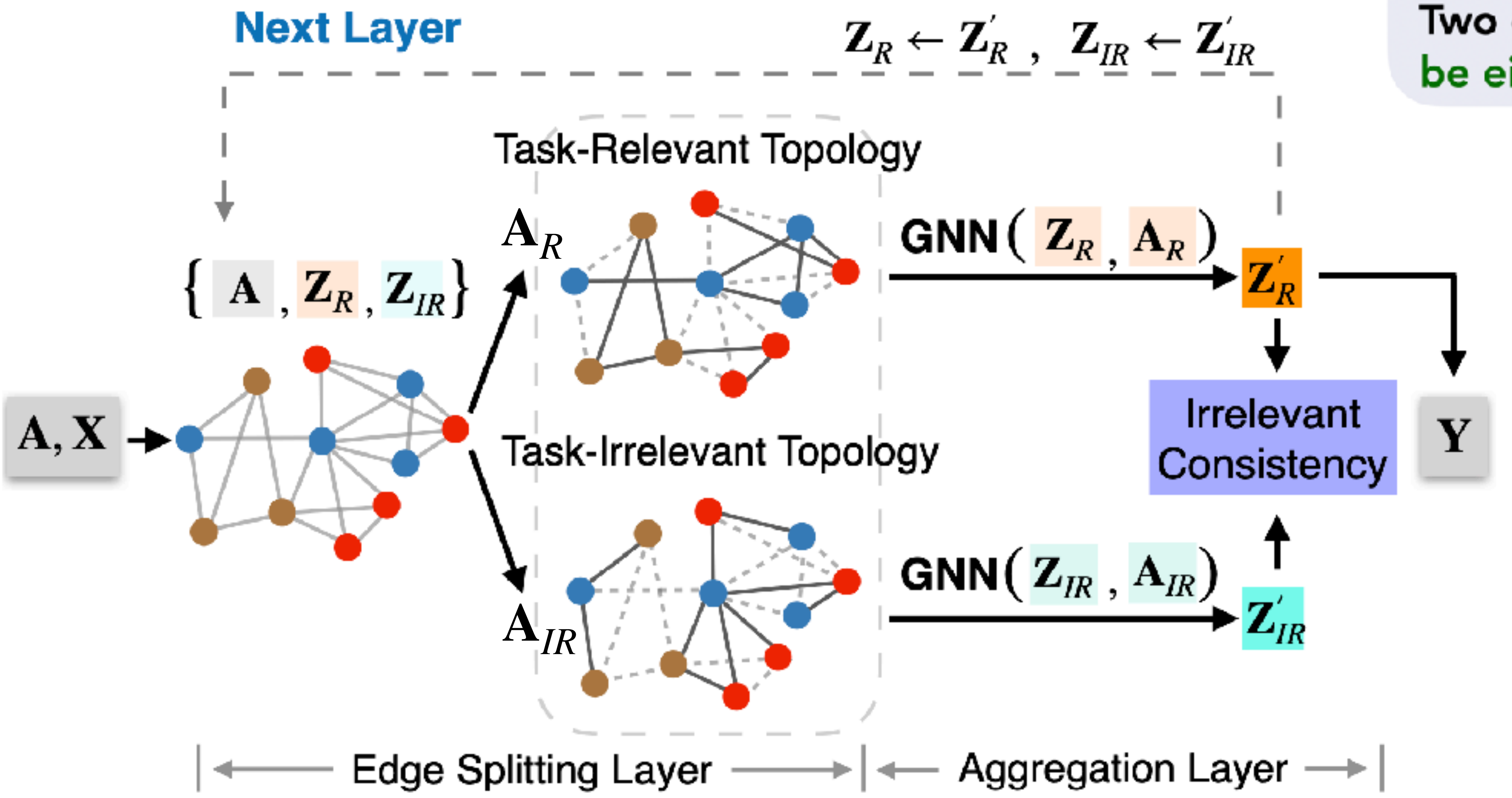
Proposed Solution #2: Edge Splitting Graph Neural Networks

Conventional Smoothness Assumption

Two connected nodes mostly share task-beneficial similarity.

Disentangled Smoothness Assumption (Ours)

Two connected nodes share similarity in some features, which could be either relevant or irrelevant (even harmful) to learning tasks.

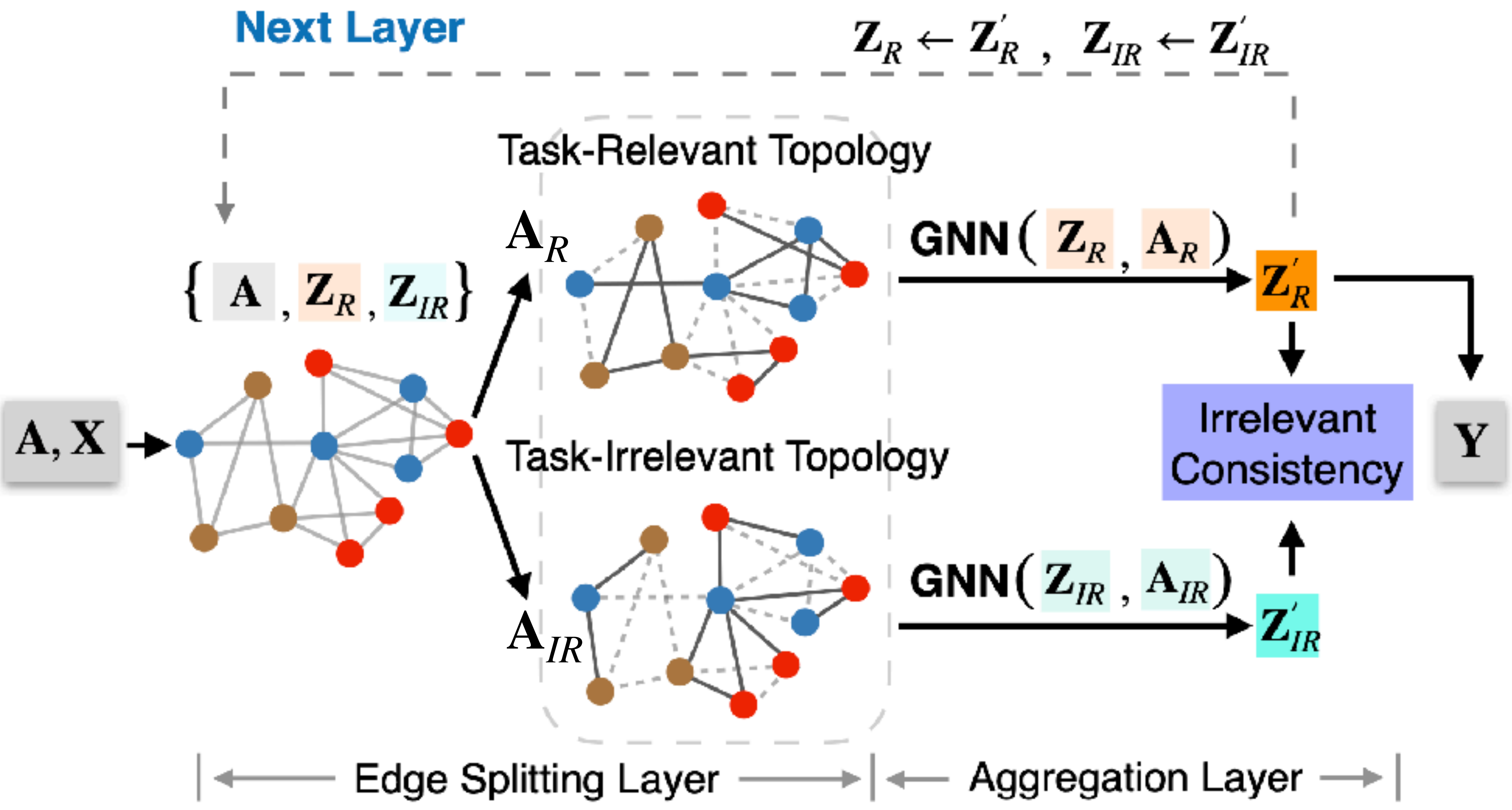


Our ES-GNN addresses heterophilic graphs by partitioning network topology, and disentangling node features.

Jingwei Guo, et.al. ES-GNN: Generalizing Graph Neural Networks Beyond Homophily with Edge Splitting. *TPAMI, 2024*.

Proposed Solution #2: Edge Splitting Graph Neural Networks

Key Components



- **Residual Scoring Mechanism**

$$\begin{cases} \mathbf{A}_{\mathrm{R}(i,j)} - \mathbf{A}_{\mathrm{IR}(i,j)} = \alpha_{i,j} \\ \mathbf{A}_{\mathrm{R}(i,j)} + \mathbf{A}_{\mathrm{IR}(i,j)} = 1 \end{cases}$$

This gives us $\mathbf{A}_{\mathrm{R}(i,j)} = \frac{1+\alpha_{i,j}}{2}$ and $\mathbf{A}_{\mathrm{IR}(i,j)} = \frac{1-\alpha_{i,j}}{2}$ with $-1 \leq \alpha_{i,j} \leq 1$. To effectively quantify the interaction (or

$$\alpha_{i,j} = \tanh(\mathbf{g}\left[\mathbf{Z}_{\mathrm{R}[i,:]} \oplus \mathbf{Z}_{\mathrm{IR}[i,:]} \oplus \mathbf{Z}_{\mathrm{R}[j,:]} \oplus \mathbf{Z}_{\mathrm{IR}[j,:]}\right]^T)$$

- **Irrelevant Consistency Reg.**

// Prediction.

$\hat{\mathbf{y}}_i = \mathrm{softmax}(\mathbf{W}_F^T \mathbf{Z}_{\mathrm{R}[i,:]}^{(K)} + \mathbf{b}_F), \forall v_i \in \mathcal{V}.$

// Optimization with Irrelevant Consistency Regularization.

$\mathcal{L}_{\mathrm{ICR}} = \sum_{(v_i,v_j)\in\mathcal{E}} (1 - \delta(\hat{\mathbf{y}}_i, \hat{\mathbf{y}}_j)) \|\mathbf{Z}_{\mathrm{IR}[i,:]} - \mathbf{Z}_{\mathrm{IR}[j,:]}\|_2^2.$

$\mathcal{L}_{\mathrm{pred}} = -\frac{1}{|\mathcal{V}_{\mathrm{trn}}|} \sum_{i\in\mathcal{V}_{\mathrm{trn}}} \mathbf{y}_i^T \log(\hat{\mathbf{y}}_i).$

Minimize $\mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{ICR}} \mathcal{L}_{\mathrm{ICR}}.$

Jingwei Guo, et.al. ES-GNN: Generalizing Graph Neural Networks Beyond Homophily with Edge Splitting. *TPAMI, 2024*.

Proposed Solution #2: Edge Splitting Graph Neural Networks

Theoretical Justification

- Conventional GNNs

Graph Denoising Problem

$$\arg\min_{\mathbf{Z}} \ \|\mathbf{Z}-\mathbf{X}\|_2^2 + \xi \cdot tr(\mathbf{Z}^T\mathbf{L}\mathbf{Z})$$

Our Analysis ↓ $L = L_R + L_{IR}$

$$\arg\min_{\mathbf{Z}} \ \|\mathbf{Z}-\mathbf{X}\|_2^2 + \xi \cdot tr(\mathbf{Z}^T\mathbf{L}_R\mathbf{Z}) + \xi \cdot tr(\mathbf{Z}^T\mathbf{L}_{IR}\mathbf{Z})$$

Possible classification-harmful information could be preserved in $\mathbf{Z}$

- Our ES-GNN

Disentangled Graph Denoising Problem

$$\arg\min_{\mathbf{Z}_R,\mathbf{Z}_{IR}} \ \|\mathbf{Z}_R-\mathbf{X}_{IR}\|_2^2 + \|\mathbf{Z}_{IR}-\mathbf{X}_{IR}\|_2^2 + \xi \cdot tr(\mathbf{Z}_R^T\mathbf{L}_R\mathbf{Z}_R) + \xi \cdot tr(\mathbf{Z}_{IR}^T\mathbf{L}_{IR}\mathbf{Z}_{IR})$$

where $\mathbf{L}_R = \mathbf{D}_R - \mathbf{A}_R, \mathbf{L}_{IR} = \mathbf{D}_{IR} - \mathbf{A}_{IR}$

s.t. $\mathbf{A}_R + \mathbf{A}_{IR} = \mathbf{A}$

$\mathbf{A}_{R(i,j)}, \mathbf{A}_{IR(i,j)} \in [0, 1].$

Possible classification-harmful information can be excluded from $\mathbf{Z}_R$ & disentangled in $\mathbf{Z}_{IR}$

Jingwei Guo, et.al. ES-GNN: Generalizing Graph Neural Networks Beyond Homophily with Edge Splitting. *TPAMI, 2024*.

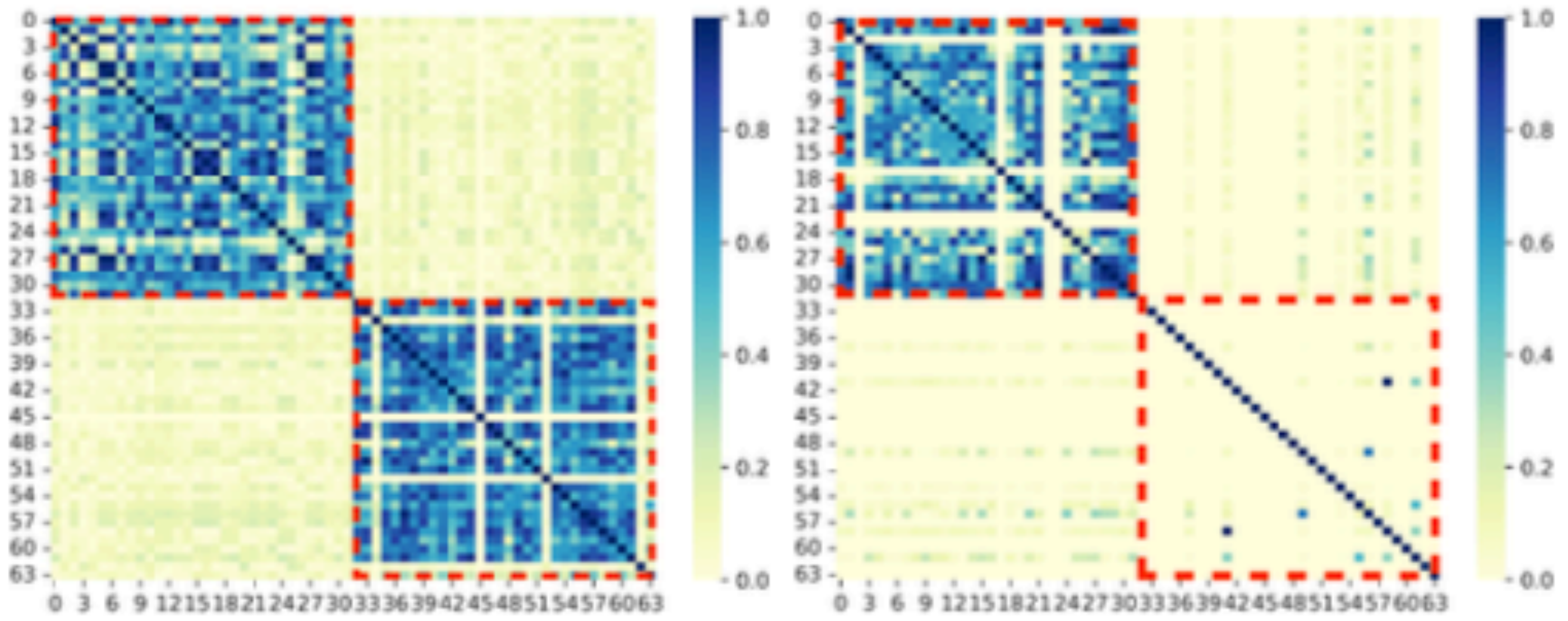
Proposed Solution #2: Edge Splitting Graph Neural Networks

Empirical Results

Node classification accuracies (%) over 100 runs. Error Reduction gives the average improvement of ES-GNN upon baselines w/o Basic GNNs.

Datasets	Heterophilic Graphs							Homophilic Graphs			
	Squirrel	Chameleon	Wisconsin	Cornell	Texas	Twitch-DE	Actor	Cora	Citeseer	Pubmed	Polblogs
GCN [30]	55.2±1.5	67.6±2.0	59.5±3.6	52.8±6.0	61.7±3.7	74.0±1.2	31.2±1.3	79.7±1.2	69.5±1.7	78.7±1.6	89.4±0.9
SGC [6]	50.7±1.3	61.9±2.6	53.7±3.9	51.2±0.9	51.4±2.2	73.9±1.3	30.9±0.6	79.1±1.0	69.9±2.0	76.6±1.3	89.0±1.5
GAT [26]	54.8±2.2	67.3±2.2	57.9±4.5	50.4±5.9	55.4±5.9	73.7±1.3	30.5±1.2	82.0±1.1	69.9±1.7	78.6±2.0	87.4±1.1
NeuralSparse [48]	40.0±1.6	60.5±2.0	70.8±3.4	64.1±5.5	66.4±5.7	71.3±1.3	35.5±1.1	78.5±1.4	69.7±1.8	79.1±1.2	89.3±0.9
GCN-LPA [49]	54.2±1.1	63.4±1.9	63.3±3.7	65.6±7.3	61.2±7.6	74.0±1.2	37.8±0.9	80.4±1.5	69.7±1.7	79.7±1.3	**89.7±0.8**
DisenGCN [66]	42.4±1.6	58.4±2.3	78.1±4.0	77.4±4.4	71.3±5.7	73.5±1.7	36.7±1.2	81.5±1.3	69.2±1.7	80.0±1.6	89.5±0.9
FactorGCN [33]	56.6±2.4	69.8±2.0	64.2±4.8	50.6±1.8	69.5±6.5	73.1±1.4	29.0±1.4	75.2±1.6	61.6±2.0	72.9±2.3	87.9±1.7
VEPM [71]	50.3±1.7	67.3±2.1	55.6±4.9	51.2±7.0	55.8±4.3	73.3±1.2	29.3±1.1	82.2±1.2	69.1±1.9	78.8±1.6	89.5±0.9
DisCNN [72]	55.1±4.8	68.2±1.9	54.6±5.4	52.0±5.7	60.6±3.9	69.2±0.8	30.2±1.3	78.2±1.4	66.2±2.2	77.6±1.7	89.6±0.9
GEN [13]	36.0±4.0	57.6±3.1	83.3±3.6	81.0±3.9	78.3±8.0	74.1±1.4	37.3±1.4	79.8±1.3	69.7±1.6	78.9±1.7	89.6±1.4
WRGAT [14]	39.6±1.4	57.7±1.6	82.9±4.5	79.2±3.5	80.5±6.1	70.0±1.3	38.6±1.1	71.7±1.5	64.1±1.9	73.3±2.1	88.2±1.2
H2GCN [10]	45.1±1.9	62.9±1.9	82.6±4.0	79.6±4.9	79.8±7.3	73.1±1.5	38.4±1.0	81.4±1.4	68.7±2.0	78.0±2.0	89.0±1.0
FAGCN [18]	50.4±2.6	68.9±1.8	82.3±4.4	79.4±5.5	80.3±5.5	74.1±1.4	37.9±1.0	82.6±1.3	70.3±1.6	80.0±1.7	89.3±1.1
GPR-GNN [11]	54.1±1.6	69.6±1.7	82.7±4.1	79.9±5.3	81.7±4.9	74.0±1.6	38.0±1.1	81.5±1.5	69.6±1.7	79.8±1.3	89.5±0.8
GloGNN++ [23]	63.3±1.2	71.4±2.0	84.9±4.2	82.0±3.5	81.4±5.6	72.8±1.1	38.2±1.2	80.9±1.4	70.5±1.9	76.8±2.1	89.6±0.8
ACM-GCN [46]	**67.0±1.3**	**75.3±2.2**	84.3±4.5	82.1±4.9	82.2±5.9	74.2±0.9	36.6±1.0	81.3±1.0	69.4±1.7	79.5±1.4	89.6±0.9
GOAL [47]	57.9±0.9	71.3±2.0	70.5±5.1	54.9±6.6	72.0±7.4	68.5±1.5	36.3±1.0	80.6±1.4	69.7±2.0	78.7±1.3	88.7±1.6
ES-GNN (ours)	62.4±1.4	72.3±2.1	**85.3±4.6**	**82.2±4.0**	**82.3±5.7**	**74.7±1.1**	**38.9±0.8**	**83.0±1.1**	**70.7±1.7**	**80.7±1.4**	**89.7±0.9**
Error Reduction	**11.5%**	**6.4%**	**11.0%**	**11.7%**	**9.4%**	**2.2%**	**3.2%**	**3.3%**	**2.3%**	**2.6%**	**0.5%**

Heterophilic vs. Homophilic



(a) Chameleon (b) Cora

Block #1: Task-Relevant

Block #2: Task-Irrelevant

ES-GNN disentangle task-relevant and irrelevant features, showcasing universal adaptivity on different network types.

Jingwei Guo, et.al. ES-GNN: Generalizing Graph Neural Networks Beyond Homophily with Edge Splitting. *TPAMI, 2024*.

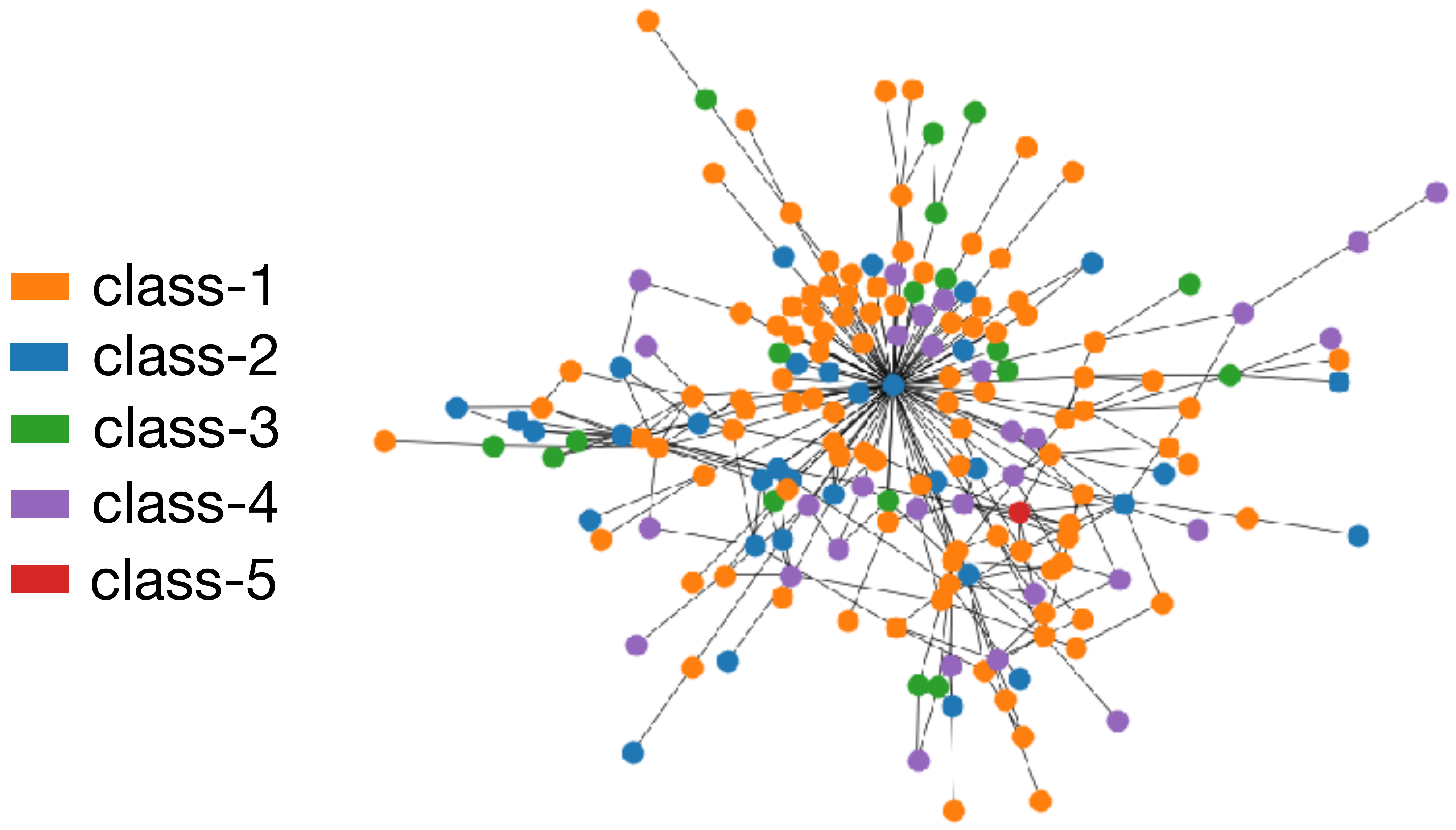
Solution #3: Diverse Spectral Filtering Framework

Jingwei Guo, et.al. Graph Neural Networks with Diverse Spectral Filtering. *WWW, 2023*.

Proposed Solution #3: Diverse Spectral Filtering

New Graph Heterophily Insights From a Broader Context

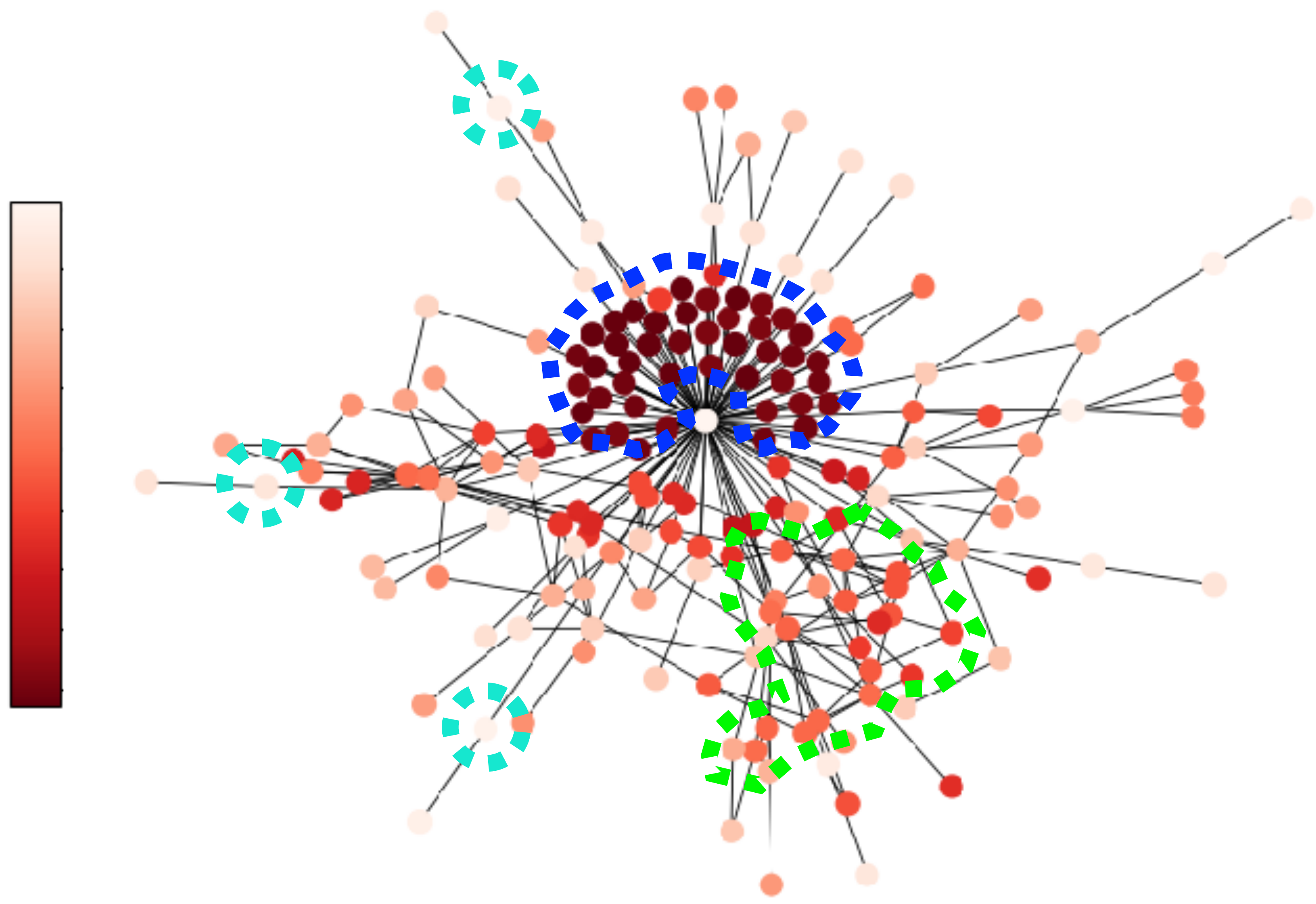
- Edge-level



label similarity / dissimilarity

Pairwise Distinction

- Subgraph-level



structure similarity / dissimilarity

Regional Disparity

Jingwei Guo, et.al. Graph Neural Networks with Diverse Spectral Filtering. *WWW, 2023.*

Proposed Solution #3: Diverse Spectral Filtering

$$\mathbf{Z} = g_{\psi}(\hat{\mathbf{L}})\mathbf{X} = \sum_{k=0}^{K} \psi_k P_k(\hat{\mathbf{L}})\mathbf{X}$$

Filter Function

Polynomial Approx. with Shared Parameters for each K-hop Neighborhood

Most spectral GNNs assumes homogenous distributions between different graph regions. ☹

$$\mathbf{Z} = \sum_{k=0}^{K} \begin{pmatrix} \beta_{k,1} & 0 & \cdots & 0 \\ 0 & \beta_{k,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \beta_{k,N} \end{pmatrix} P_k(\hat{\mathbf{L}})\mathbf{X}$$

We augment the original parameters into node-specific filter weights to model diverse regional patterns. ☺

Jingwei Guo, et.al. Graph Neural Networks with Diverse Spectral Filtering. *WWW, 2023*.

Proposed Solution #3: Diverse Spectral Filtering

Label Homophily

$$h = \frac{|\{(v_i,v_j)|\mathbf{y}_i=\mathbf{y}_j \wedge (v_i,v_j)\in\mathcal{E}\}|}{|\mathcal{E}|}$$

[0,1]

→

Definition 1 (Local Label Homophily). We define the Local Label Homophily as a measure of the local homophily level surrounding each node v_i:

$$h_i = \frac{|\{(v_p,v_q)|\mathbf{y}_p = \mathbf{y}_q \wedge (v_p,v_q) \in \mathcal{E}_{i,k}\}|}{|\mathcal{E}_{i,k}|}$$

Here, h_i directly computes the edge homophily ratio [50] on the subgraph made up of the k-hop neighbors, and $\mathcal{E}_{i,k} = \{(v_p,v_q)|v_p,v_q \in \mathcal{N}_{i,k} \wedge (v_p,v_q) \in \mathcal{E}\}$ denotes its edge set.

Graph Frequency (Eigenvalue)

$$\lambda_n = \mathbf{u}_n^T \hat{\mathbf{L}} \mathbf{u}_n = \sum_{(v_p,v_q)\in\mathcal{E}} \left(\frac{1}{\sqrt{\deg_p}}\mathbf{u}_{n,p} - \frac{1}{\sqrt{\deg_q}}\mathbf{u}_{n,q}\right)^2$$

[0,1]

→

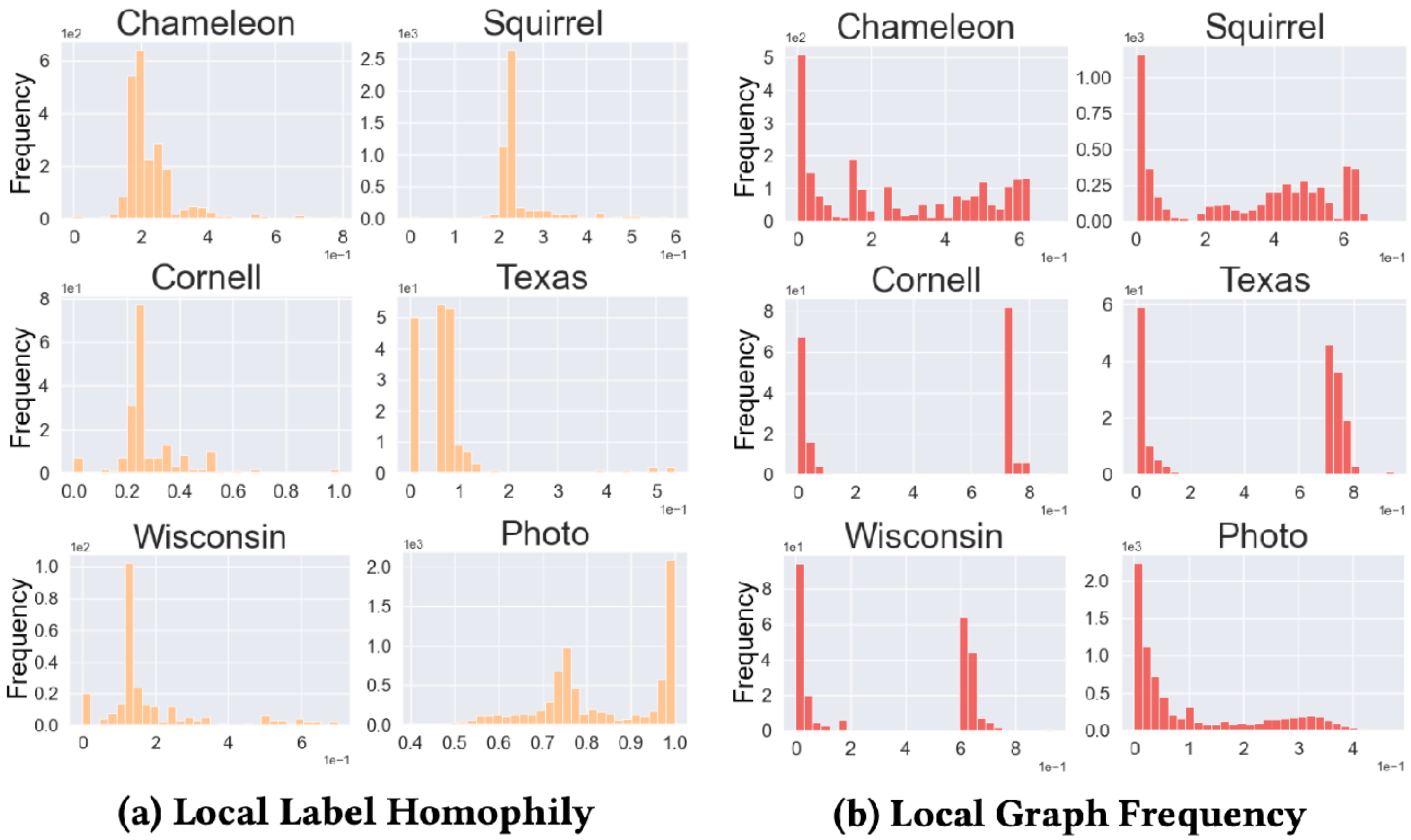
Definition 2 (Local Graph Frequency). The Local Graph Frequency is defined by measuring the local smoothness level of the decomposed Laplacian eigenbases, and for each node v_i we have:

$$\lambda_{n,i} = \sum_{(v_p,v_q)\in\mathcal{E}_{i,k}} \left(\frac{1}{\sqrt{\deg_p}}\mathbf{u}_{n,p} - \frac{1}{\sqrt{\deg_q}}\mathbf{u}_{n,q}\right)^2$$

where $\lambda_{n,i}$ denotes the frequency or smoothness level of each Laplacian eigenbasis $\mathbf{u}_n$ upon the subgraph induced by the k-hop neighbors. Since all summed elements in Eq. 1 are positive and $\mathcal{E}_{i,k} \subseteq \mathcal{E}$, we can always have a $\xi_i \in (0,1)$ such that $\lambda_{n,i} = \xi_i \lambda_n$.

Jingwei Guo, et.al. Graph Neural Networks with Diverse Spectral Filtering. *WWW, 2023*.

Proposed Solution #3: Diverse Spectral Filtering



Evident regional disparity can be observed on Real-world Graphs.

Jingwei Guo, et.al. Graph Neural Networks with Diverse Spectral Filtering. *WWW, 2023*.

Proposed Solution #3: Diverse Spectral Filtering

Reshaping Homogenous Spectral Filtering

Scalar coefficient ↓ Graph frequency ↓

$$\mathbf{Z} = \sum_{n=1}^{N} \tilde{S}_n \cdot \mathbf{U}_n \qquad \tilde{S}_n = \sum_{k=0}^{K} \alpha_k P_k(\lambda_n) \mathbf{U}_n^T \mathbf{X}$$

Hadamard product ↓ The i-th element of vector coefficients ↓

$$\mathbf{Z} = \sum_{n=1}^{N} \tilde{\mathbf{S}}_n \odot \mathbf{U}_n \qquad \tilde{\mathbf{S}}_n(i) = \sum_{k=0}^{K} \alpha_k P_k(\lambda_{n,i}) \mathbf{U}_n^T \mathbf{X}$$

↑ Vector ↑ Local graph frequency

Jingwei Guo, et.al. Graph Neural Networks with Diverse Spectral Filtering. *WWW, 2023.*

$\mathbf{X}$ is taken as one-dimension as an example

Proposed Solution #3: Diverse Spectral Filtering

Approximation Trick

<u>Substitution</u> using $\lambda_{n,i} = \xi_i \lambda_n$ s.t. $0 < \xi_i < 1$

Proposition 1. Suppose a K-order polynomial function $f : [0, 2] \to \mathbb{R}$ with polynomial basis $P_k(\cdot)$ and coefficients $\{\alpha_k\}_{k=0}^{K}$ in real number. For any pair of variables $x, \hat{x} \in [0, 2]$ satisfying $x = \xi \hat{x}$ where ξ is a constant real number, we always have a function $g : [0, 2] \to \mathbb{R}$ with the same polynomial basis but a different set of coefficients $\{\beta_k\}_{k=0}^{K}$ such that $f(x) = g(\hat{x})$.

It allows $\mathbf{S}'_n(i) = \sum_{k=0}^{K} \alpha_k P_k(\lambda_{n,i}) \mathrm{U}_n^T \mathbf{X} = \sum_{k=0}^{K} \beta_{k,i} P_k(\lambda_i) \mathrm{U}_n^T \mathbf{X}$

$$\mathbf{Z} = \sum_{k=0}^{K} \textbf{diag}(\beta_{k,1}, \beta_{k,2}, \cdots, \beta_{k,N}) P_k(\hat{\mathbf{L}}) \mathbf{X}$$

Jingwei Guo, et.al. Graph Neural Networks with Diverse Spectral Filtering. *WWW, 2023.*

Proposed Solution #3: Diverse Spectral Filtering

Key Designs

- Local and Global Weight Decomposition

$$\beta_{k,i} \leftarrow \gamma_i \cdot \alpha_{k,i}$$

γ_i: global invariant graph properties

$\alpha_{k,i}$: local diverse node contexts

- Position-aware Filter Weights

$$\arg\min_{\mathbf{P}} \|\mathbf{P}^{(0)} - \mathbf{P}\|_2^2 + \kappa_1 tr(\mathbf{P}^T\hat{\mathbf{L}}\mathbf{P}) + \kappa_2\|\mathbf{P}^T\mathbf{P} - \mathbf{I}\|_2^2$$

$\mathbf{P}$ denotes node positional embeddings

$$\alpha_{k,i} = \sigma_p(\mathbf{W}^{(k)}\mathbf{P}_i^{(k)} + \mathbf{b}^{(k)}) \qquad k = 1, 2, \cdots, K$$

Jingwei Guo, et.al. Graph Neural Networks with Diverse Spectral Filtering. *WWW, 2023*.

Proposed Solution #3: Diverse Spectral Filtering

Empirical Results — Classification Accuracy

Table 2: Node classification accuracies (%) ± 95% confidence interval over 100 runs.

Datasets	Heterophilic Graphs						Homophilic Graphs				
	Chameleon	Squirrel	Wisconsin	Cornell	Texas	Twitch-DE	Cora	Citeseer	Pubmed	Computers	Photo
GPR-GNN [9]	69.01±0.50	55.39±0.33	82.72±0.85	80.81±0.78	81.66±1.02	74.07±0.18	89.03±0.20	77.63±0.28	90.10±0.44	92.34±0.13	95.34±0.09
DSF-GPR-I	71.18±0.52	57.08±0.29	**87.64**±0.79	84.76±0.90	85.44±1.05	74.58±0.16	**89.64**±0.20	78.03±0.26	90.26±0.08	92.49±0.12	95.64±0.07
DSF-GPR-R	**71.64**±0.55	**58.44**±0.30	87.43±0.74	**84.93**±0.90	**85.56**±0.93	**74.81**±0.14	89.63±0.17	**78.22**±0.29	**90.51**±0.07	**92.80**±0.12	**95.73**±0.08
Our Improv.	2.63%	3.05%	4.92%	4.12%	3.9%	0.74%	0.61%	0.59%	0.41%	0.46%	0.39%
BernNet [20]	70.59±0.42	56.63±0.32	85.00±0.94	82.10±0.95	82.20±0.98	74.45±0.15	88.72±0.23	77.52±0.29	90.21±0.46	92.57±0.10	95.42±0.08
DSF-Bern-I	72.95±0.53	59.45±0.32	**88.23**±0.81	**85.07**±0.93	**84.59**±1.07	74.96±0.15	89.05±0.22	**78.32**±0.27	90.40±0.10	92.76±0.10	95.73±0.07
DSF-Bern-R	**73.60**±0.53	**59.99**±0.30	88.02±0.91	84.29±0.93	84.42±1.00	**75.00**±0.15	**89.10**±0.22	78.27±0.26	**90.52**±0.10	**92.84**±0.10	**95.79**±0.06
Our Improv.	3.01%	3.36%	3.23%	2.97%	2.39%	0.55%	0.38%	0.80%	0.31%	0.27%	0.37%
JacobiConv [42]	73.71±0.42	57.22±0.24	83.21±0.68	82.34±0.88	82.42±0.90	74.34±0.12	89.24±0.19	77.81±0.29	89.50±0.47	92.26±0.10	95.62±0.06
DSF-Jacobi-I	74.88±0.39	58.26±0.26	85.34±0.74	**84.54**±0.81	83.68±1.12	74.65±0.13	89.54±0.19	78.18±0.26	89.78±0.09	92.38±0.11	**95.76**±0.07
DSF-Jacobi-R	**75.00**±0.38	**59.23**±0.27	**86.13**±0.70	84.39±0.88	**84.46**±0.81	**74.75**±0.15	**89.66**±0.19	**78.23**±0.25	**90.07**±0.10	**92.44**±0.11	95.75±0.08
Our Improv.	1.29%	2.01%	2.92%	2.20%	2.04%	0.41%	0.42%	0.42%	0.41%	0.18%	0.14%

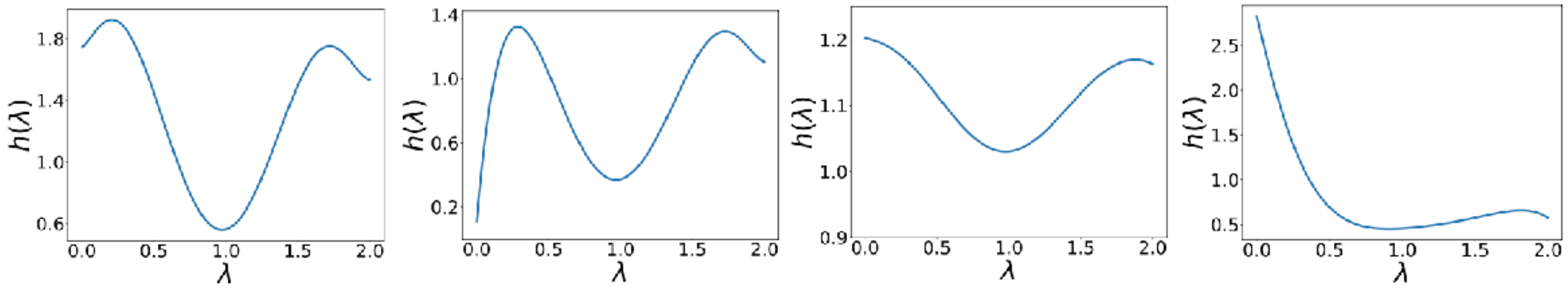
DSF can be readily plug-and-play in multiple spectral GNNs and consistently improve their performance.

Jingwei Guo, et.al. Graph Neural Networks with Diverse Spectral Filtering. *WWW, 2023.*

Proposed Solution #3: Diverse Spectral Filtering

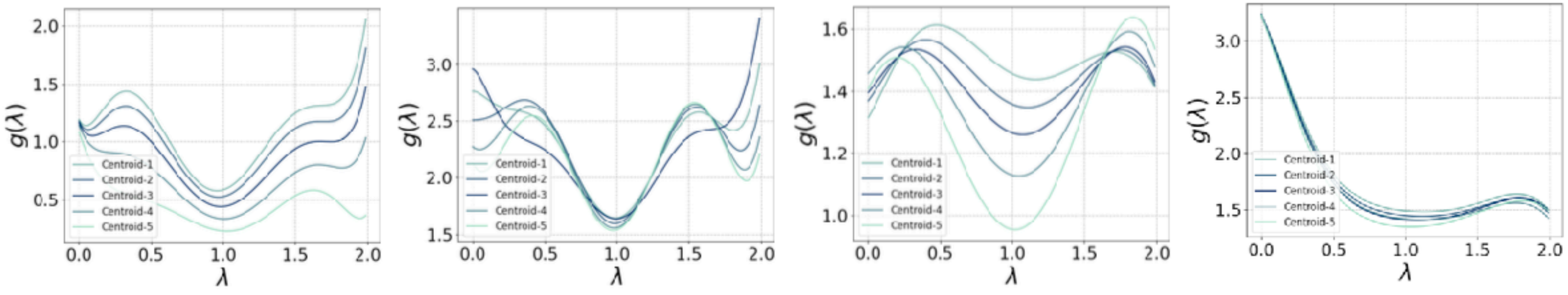
Empirical Results — Interpretable Analysis #1

- Traditional Spectral Filter (BernNet as an Example):



(f) Chameleon (h) Squirrel (j) Cornell (b) CiteSeer

- Our Spectral Filter (DSF-Bern):

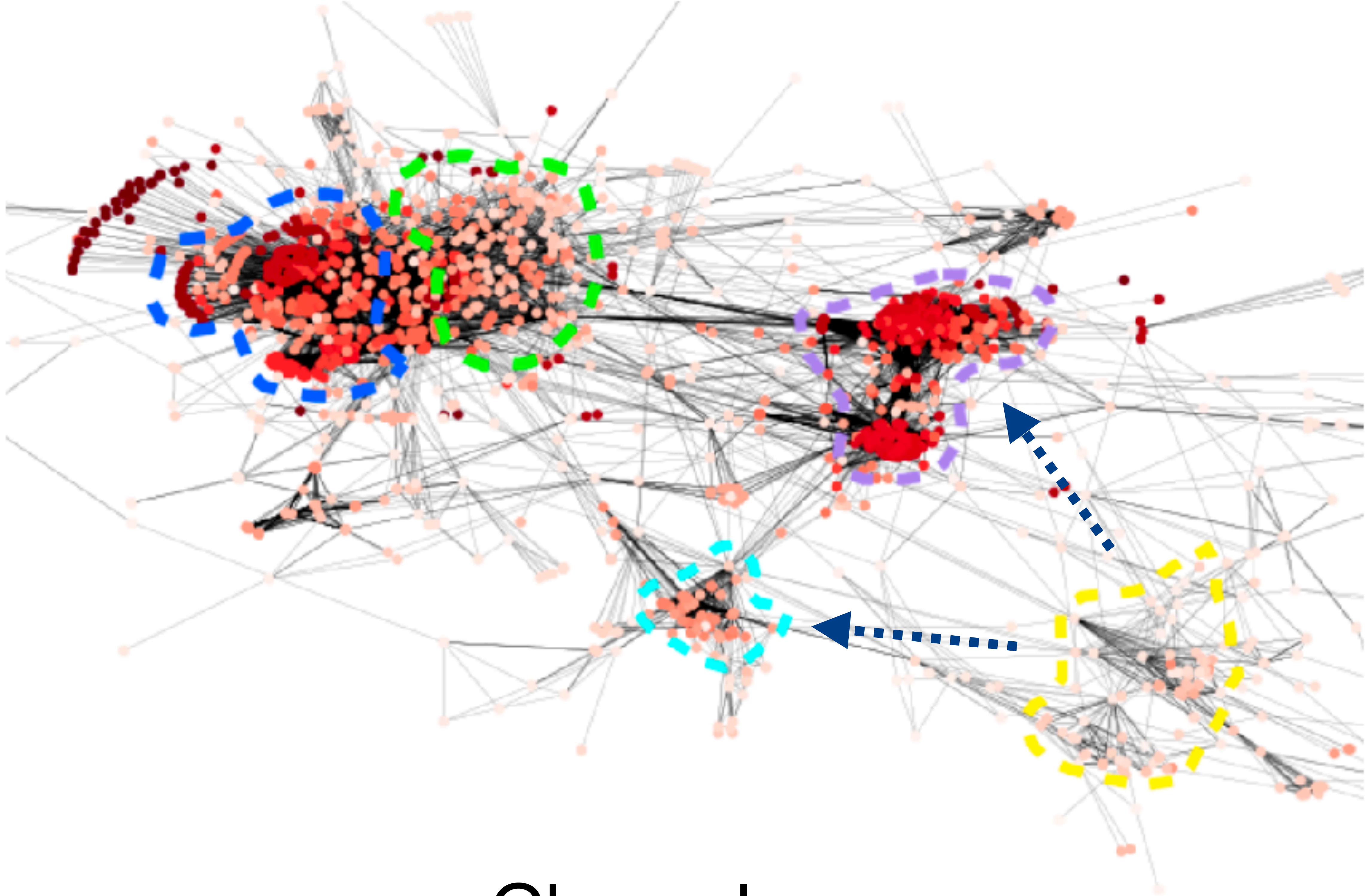


(a) Chameleon (b) Squirrel (c) Cornell (d) Citeseer

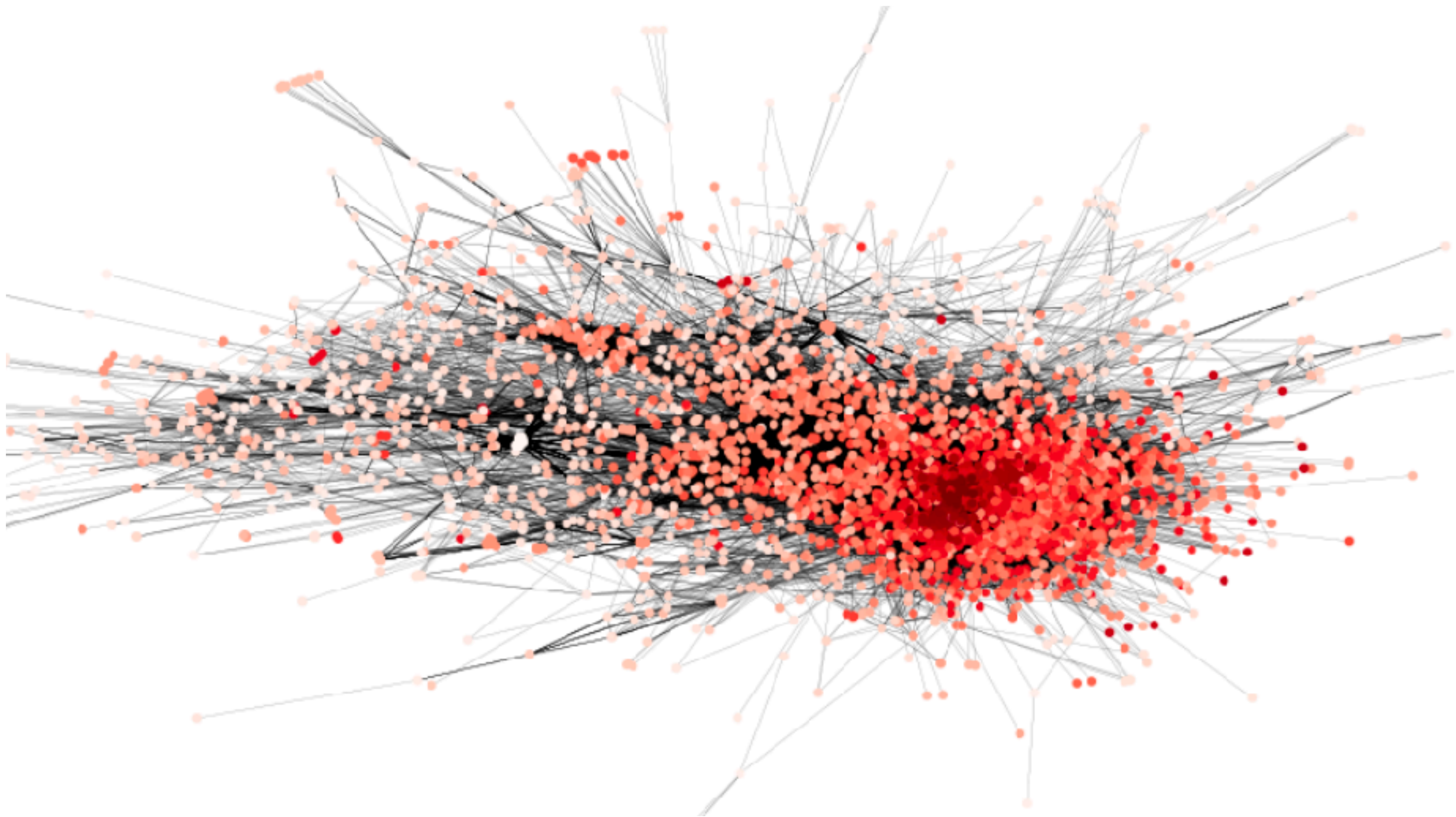
Jingwei Guo, et.al. Graph Neural Networks with Diverse Spectral Filtering. *WWW, 2023.*

Proposed Solution #3: Diverse Spectral Filtering

Empirical Results — Interpretable Analysis #2



Chameleon



Squirrel

DSF captures regional disparity with node-specific filter weights.

Jingwei Guo, et.al. Graph Neural Networks with Diverse Spectral Filtering. *WWW, 2023*.

Solution #4: Spatially Adaptive Filtering Framework

Jingwei Guo, et.al. Rethinking Spectral Graph Neural Networks with Spatially Adaptive Filtering. *TNNLS (Under 2nd Review), 2025*.

Proposed Solution #4: Spatially Adaptive Filtering

Deep Delve into Spectral GNNs

- $\mathbf{Z} = \mathbf{U} g_{\psi}(\boldsymbol{\Lambda}) \mathbf{U}^{T} \mathbf{X}$ → theoretical spectral filtering

$$= g_{\psi}(\hat{\mathbf{L}})\mathbf{X} = \sum_{k=0}^{K} \psi_k P_k(\hat{\mathbf{L}})\mathbf{X} = \sum_{k=0}^{K} \omega_k \hat{\mathbf{A}}^k \mathbf{X}$$

↓ practical spatial aggregation

- What information is essentially encoded by spectral GNNs in the spatial domain?

Jingwei Guo, et.al. Rethinking Spectral Graph Neural Networks with Spatially Adaptive Filtering. *TNNLS (Under 2nd Review), 2025*.

Proposed Solution #4: Spatially Adaptive Filtering

Cross-Domain Interplay via Generalized Graph Optimization

$$\arg\min_{\mathbf{Z}} \mathscr{L} = \alpha\|\mathbf{X}-\mathbf{Z}\|_2^2 + (1-\alpha)\mathrm{tr}(\mathbf{Z}^T\gamma_\theta(\hat{\mathbf{L}})\mathbf{Z})$$

Arbitrary Linking Patterns

- $\mathbf{Z}$ refer to node representations
- α is a trade-off coefficient
- $\gamma_\theta(\hat{\mathbf{L}}) = \mathbf{U}\gamma_\theta(\mathbf{\Lambda})\mathbf{U}^T$ determines propagation rate where $\gamma_\theta(\lambda) \geq 0$

Positive Semi-definite Constraint for Convexity Optimization

Jingwei Guo, et.al. Rethinking Spectral Graph Neural Networks with Spatially Adaptive Filtering. *TNNLS (Under 2nd Review), 2025*.

Proposed Solution #4: Spatially Adaptive Filtering

- Closed-form Solution $\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = 0$

$$\mathbf{Z}^* = (\mathbf{I} + \frac{1-\alpha}{\alpha}\gamma_\theta(\hat{\mathbf{L}}))^{-1}\mathbf{X} = g_\psi(\hat{\mathbf{L}})\mathbf{X} = \mathbf{U}g_\psi(\boldsymbol{\Lambda})\mathbf{U}^T\mathbf{X}$$

Spectral Filter as a function of $\gamma_\theta(\cdot)$: $g_\psi(\lambda) = (1 + \frac{1-\alpha}{\alpha}\gamma_\theta(\lambda))^{-1}$

- Iterative Solution $\mathbf{Z}^{(k)} = \mathbf{Z}^{(k-1)} - \frac{1}{2}\frac{\partial \mathcal{L}}{\partial \mathbf{Z}}|_{\mathbf{Z}=\mathbf{Z}^{(k-1)}}$

Spatial Aggregation

$$\mathbf{Z}^{(k)} = \alpha\mathbf{X} + (1-\alpha)\hat{\mathbf{A}}^{new}\mathbf{Z}^{(k-1)}$$

New Graph Structure: $\hat{\mathbf{A}}^{new} = \mathbf{I} - \gamma_\theta(\hat{\mathbf{L}}) = \mathbf{I} - \frac{\alpha}{1-\alpha}(g_\psi(\hat{\mathbf{L}})^{-1} - \mathbf{I})$

Jingwei Guo, et.al. Rethinking Spectral Graph Neural Networks with Spatially Adaptive Filtering. *TNNLS (Under 2nd Review), 2025*.

Proposed Solution #4: Spatially Adaptive Filtering

Desirable Properties on the Newfound Graph $\hat{\mathbf{A}}^{new}$

- Non-locality

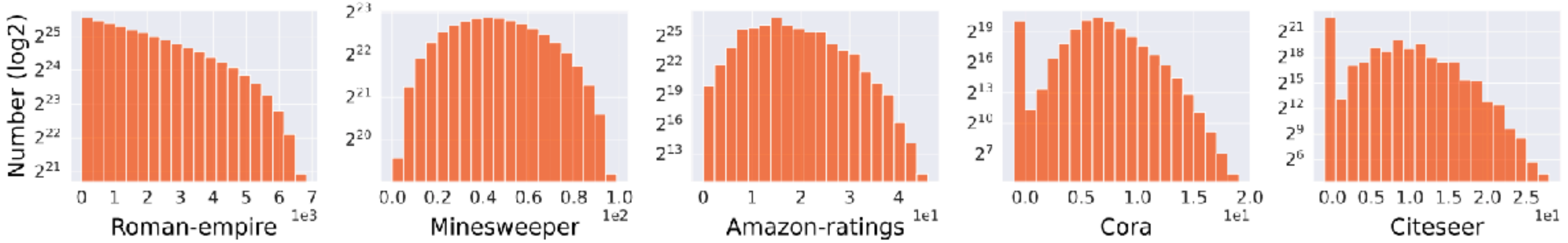


Figure 1: Distributions of connected nodes in the new graph based on their geodesic/shortest-path distance (as $\Delta_{i,j}$) in the original graph. Nodes, distant in the original graph ($\Delta_{i,j} > 1$ in x-axis), can be linked in the new graph (Number > 0 in y-axis).

- Signed Edge Weights — Global Node Label Relationships

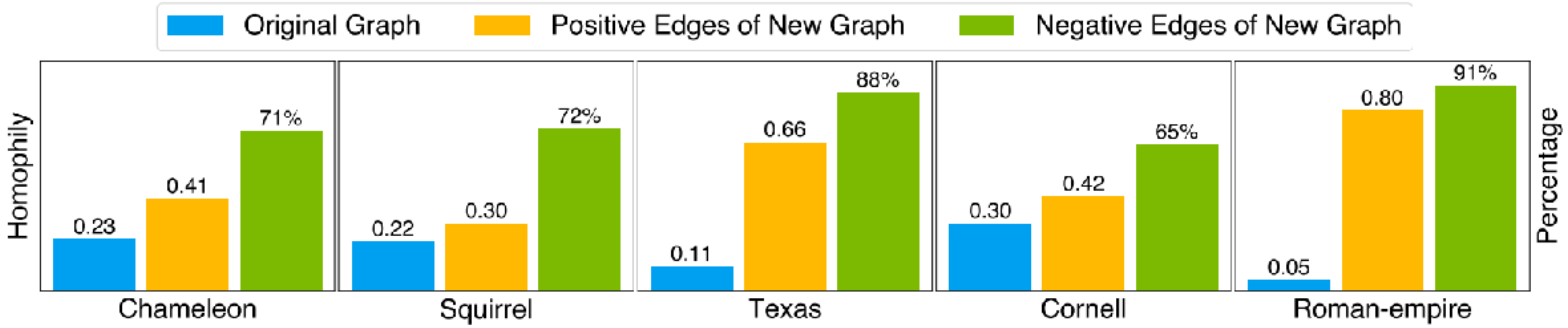


Figure 2: Left y-axis: Homophily comparison between original and new graphs, considering only positive edges (blue and yellow bars). Right y-axis: Percentage of edges connecting nodes from different classes, identified by negative edges (green bar).

Jingwei Guo, et.al. Rethinking Spectral Graph Neural Networks with Spatially Adaptive Filtering. *TNNLS (Under 2nd Review), 2025*.

Proposed Solution #4: Spatially Adaptive Filtering

Overall Framework

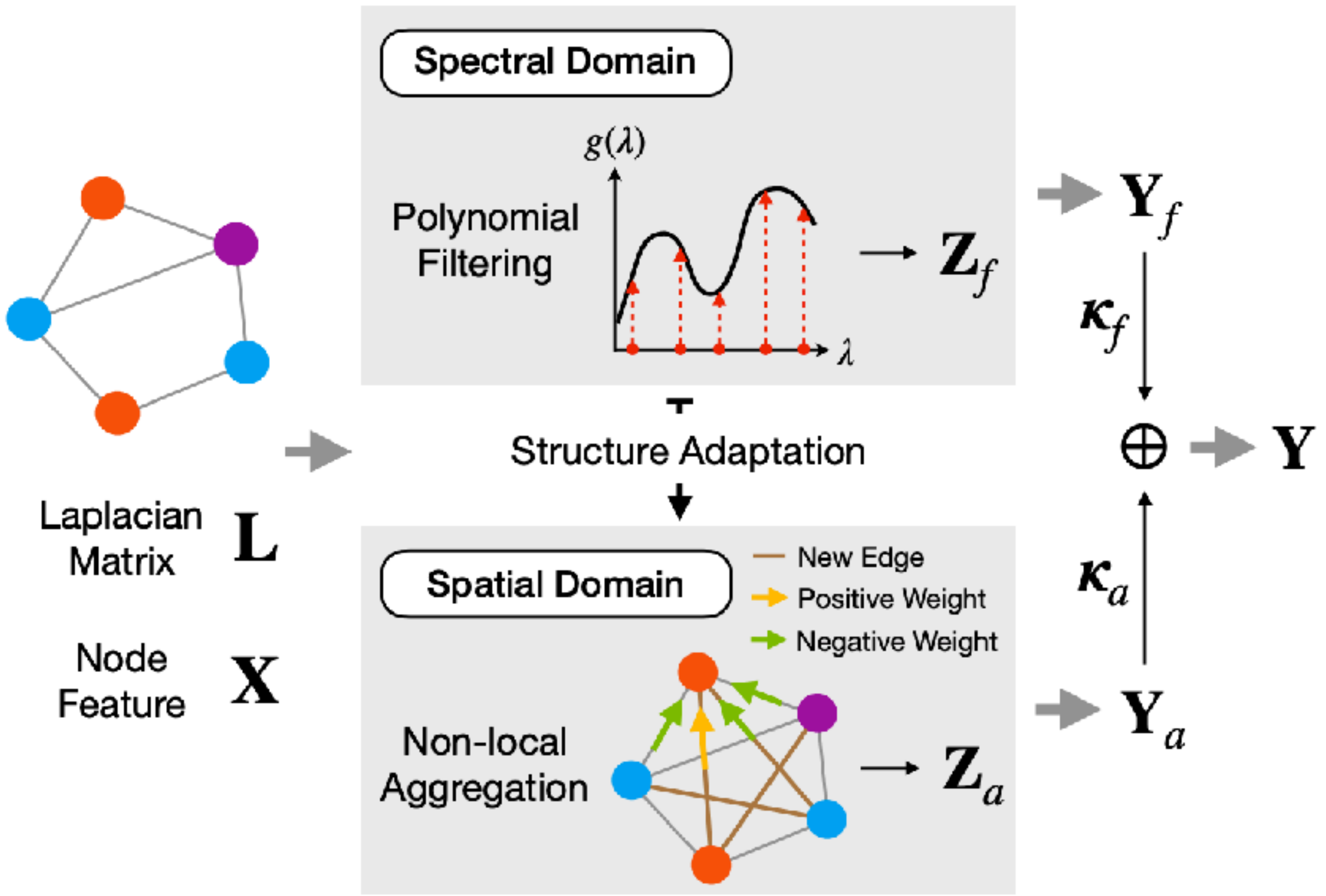


Figure 3: Illustration of the proposed SAF framework, where varying node colors represent different node labels.

SAF leverages the adapted new graph by spectral filtering for non-local aggregation with signed edge weights.

Address:

- Long-range Dependency
- Heterophilic Graph Links

Jingwei Guo, et.al. Rethinking Spectral Graph Neural Networks with Spatially Adaptive Filtering. *TNNLS (Under 2nd Review), 2025*.

Proposed Solution #4: Spatially Adaptive Filtering

Empirical Results

Table 1: Semi-supervised node classification accuracy (%) ± 95% confidence interval.

Method	Cham.	Squi.	Texas	Corn.	Actor	Cora	Cite.	Pubm.
BernNet	27.32±4.04	22.37±0.98	43.01±7.45	39.42±9.59	29.87±0.78	82.17±0.86	69.44±0.97	79.48±1.47
SAF	41.82±1.74	31.77±0.69	58.04±3.76	52.49±8.56	33.50±0.55	83.57±0.66	71.07±1.08	79.51±1.12
SAF-ϵ	41.88±2.04	32.05±0.40	**58.38±3.47**	**53.41±5.55**	**33.84±0.58**	**83.79±0.71**	**71.30±0.93**	**80.16±1.25**
Improv.	14.56%	9.68%	15.37%	13.99%	3.97%	1.62%	1.86%	0.68%

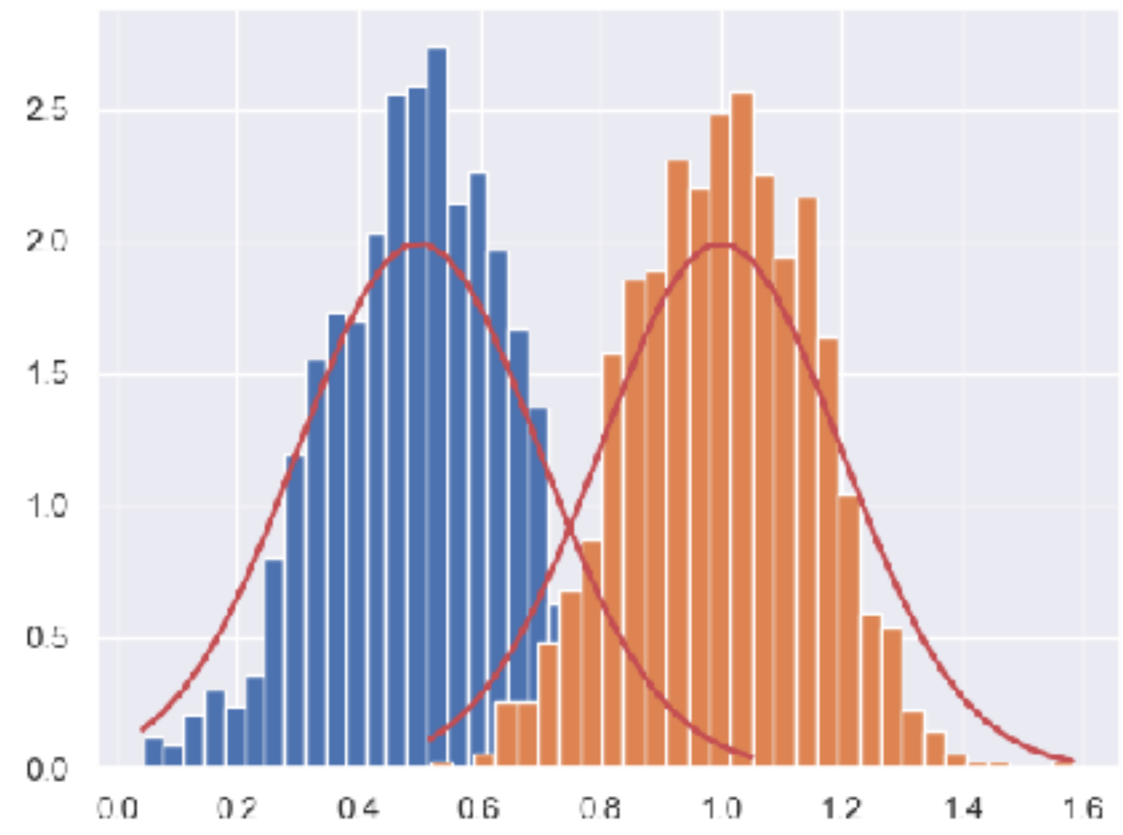
Table 2: Full-supervised node classification accuracy (%) ± 95% confidence interval.

Method	Cham.	Squi.	Texas	Corn.	Actor	Cora	Cite.	Pubm.
BernNet	68.53±1.68	51.39±0.92	92.62±1.37	92.13±1.64	41.71±1.12	88.51±0.92	80.08±0.75	88.51±0.39
SAF	**75.30±0.96**	63.63±0.81	94.10±1.48	92.95±1.97	42.93±0.79	89.80±0.69	80.61±0.81	91.49±0.29
SAF-ϵ	74.84±0.99	64.00±0.83	**94.75±1.64**	**93.28±1.80**	**42.98±0.61**	**89.87±0.51**	**81.45±0.59**	**91.52±0.30**
Improv.	6.77%	12.61%	2.13%	1.15%	1.27%	1.36%	1.37%	3.01%

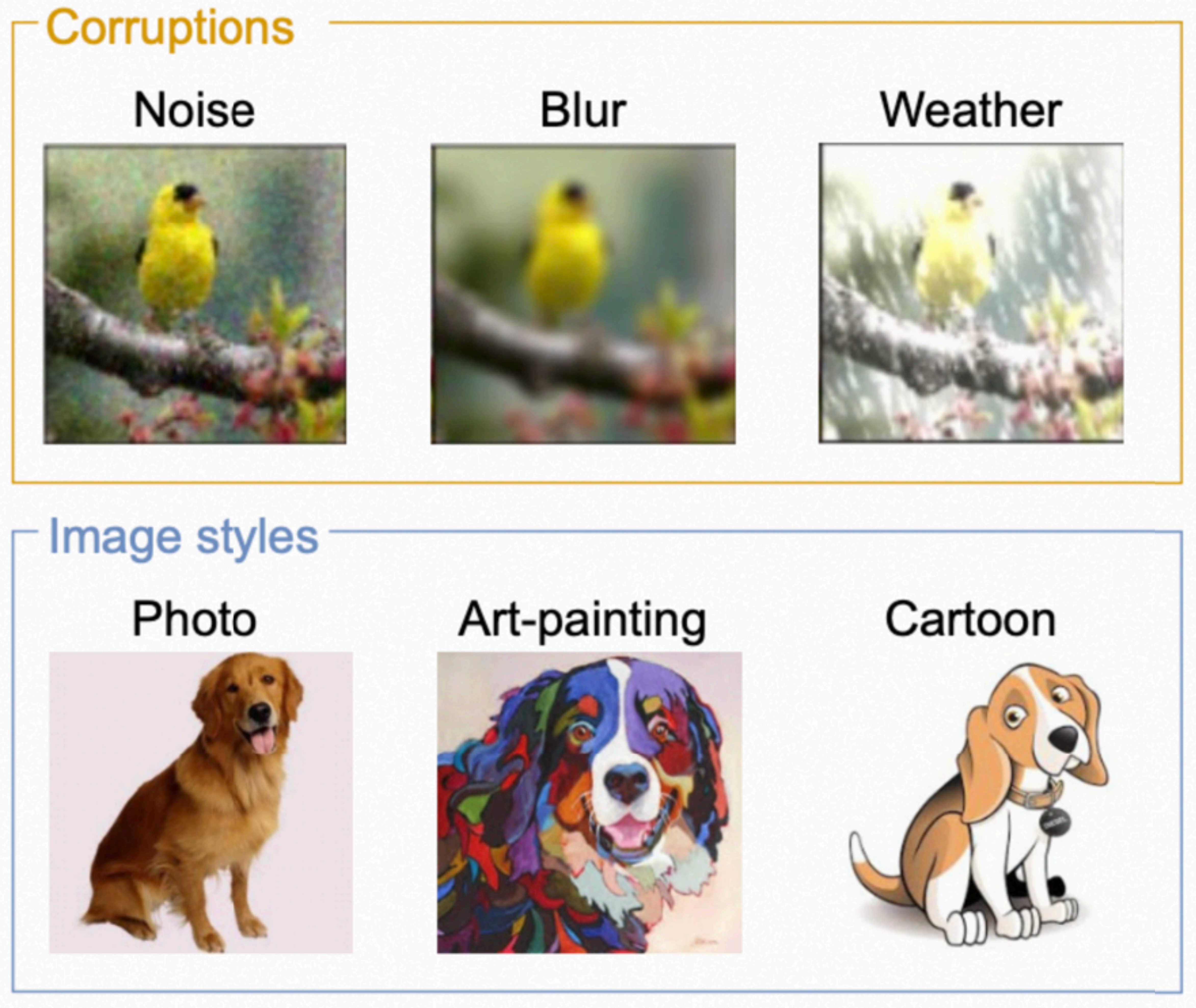
Jingwei Guo, et.al. Rethinking Spectral Graph Neural Networks with Spatially Adaptive Filtering. *TNNLS (Under 2nd Review), 2025*.

Research Project #2

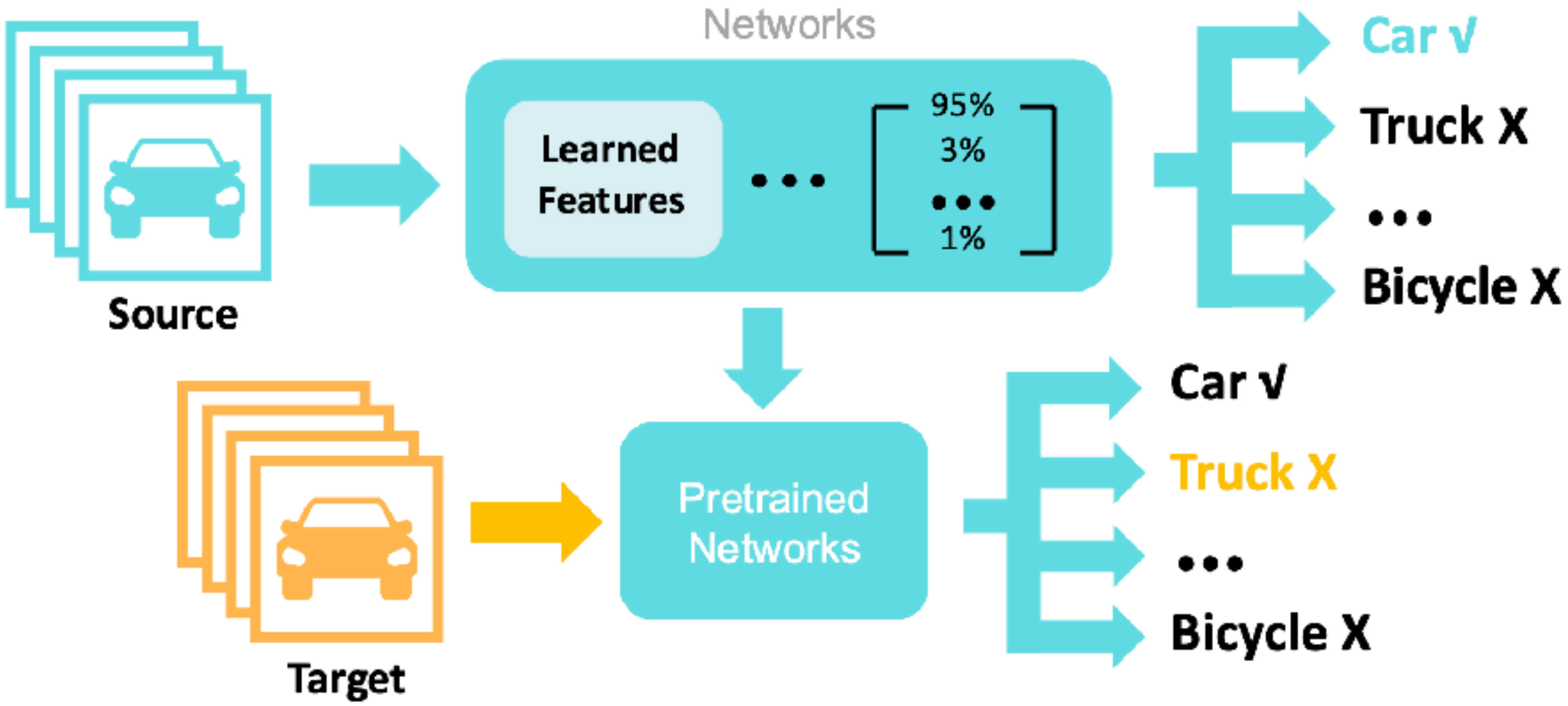
Transfer Learning Under
Distribution Shift



Research Background — Distribution / Covariate Shift



$$P_s(x) \neq P_t(x), P_s(y|x) = P_t(y|x)$$



Distribution shift may cause model performance degradation.

Lei Zhang et al. Transfer Adaptation Learning: A Decade Survey, *2022*.

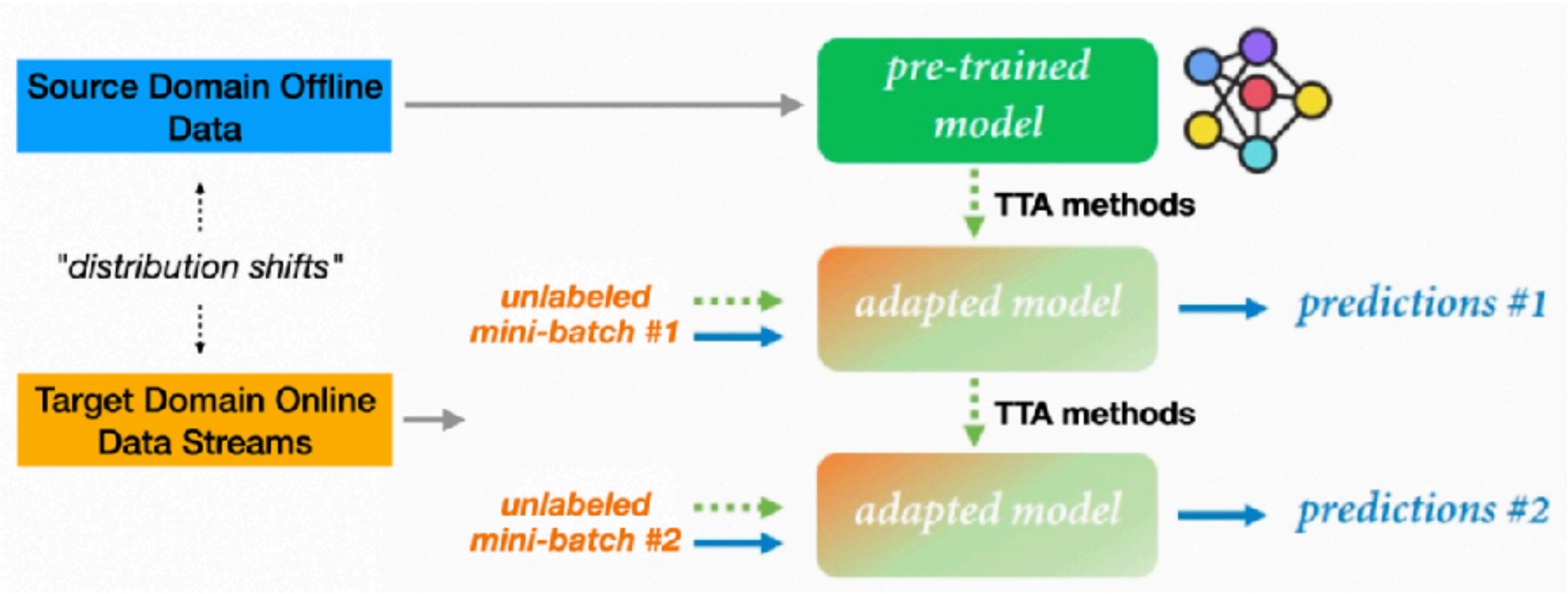
Fuzhen Zhuang et al. A Comprehensive Survey on Transfer Learning, *2020*.

Research Background — Test-time Adaptation (TTA)

TABLE I: Comparison of different transfer learning settings.

Topic	No Source Data*	No Target Labels*	Online Adaptation	Model Agnostic
Supervised Domain Adaptation	×	×	×	–
Unsupervised Domain Adaptation	×	✓	×	–
Source-free Domain Adaptation	✓	✓	×	–
Test-time Training	✓	✓	✓	×
Test-time Adaptation	✓	✓	✓	✓

Note: * denotes particular constraints during the adaptation process.

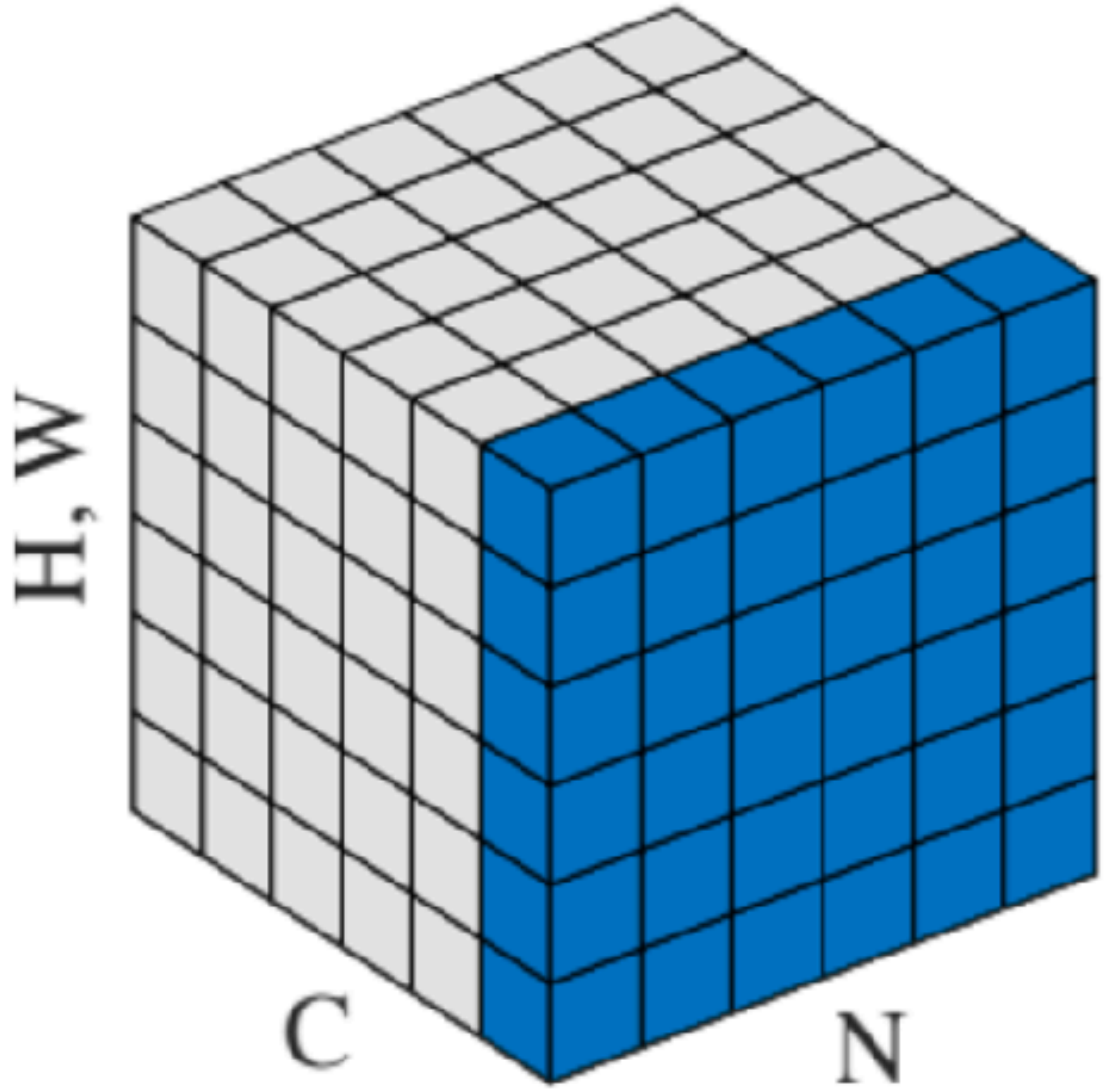


Given any pre-trained model, TTA aims to adapt it at test-time towards the unseen and unlabeled data streams.

Jian Liang, et. al. A Comprehensive Survey on Test-Time Adaptation under Distribution Shifts, *2024*.

Solution #1: Unraveling BN in TTA under Small-batch Data Streams

Batch Normalization in Domain Adaptation



$$\hat{\mathbf{x}}_i = \gamma \cdot \left(\frac{\mathbf{x}_i - \boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma}^2 + \epsilon}}\right) + \beta$$

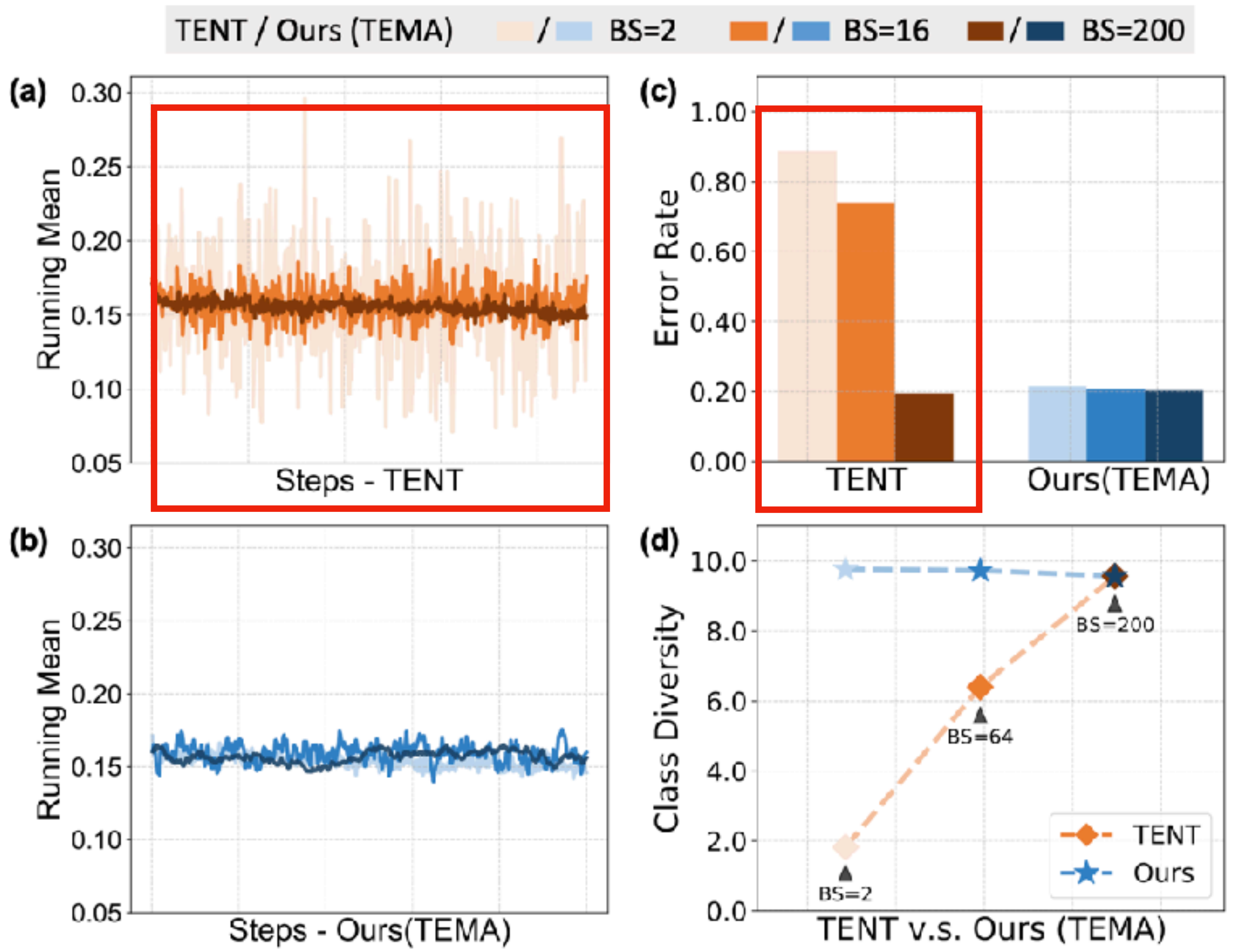
Parameters	How are they learned during training?	Testing
$\boldsymbol{\mu}, \boldsymbol{\sigma}$	exponential moving average with momentum m : $\boldsymbol{\mu} \leftarrow m\boldsymbol{\mu} + (1-m)\boldsymbol{\mu}_b$ $\boldsymbol{\sigma}^2 \leftarrow m\boldsymbol{\sigma}^2 + (1-m)\boldsymbol{\sigma}_b^2$ $\boldsymbol{\mu}_b = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i, \boldsymbol{\sigma}_b^2 = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu}_b)^2$	Fixed
γ, β	updated by gradients	Fixed

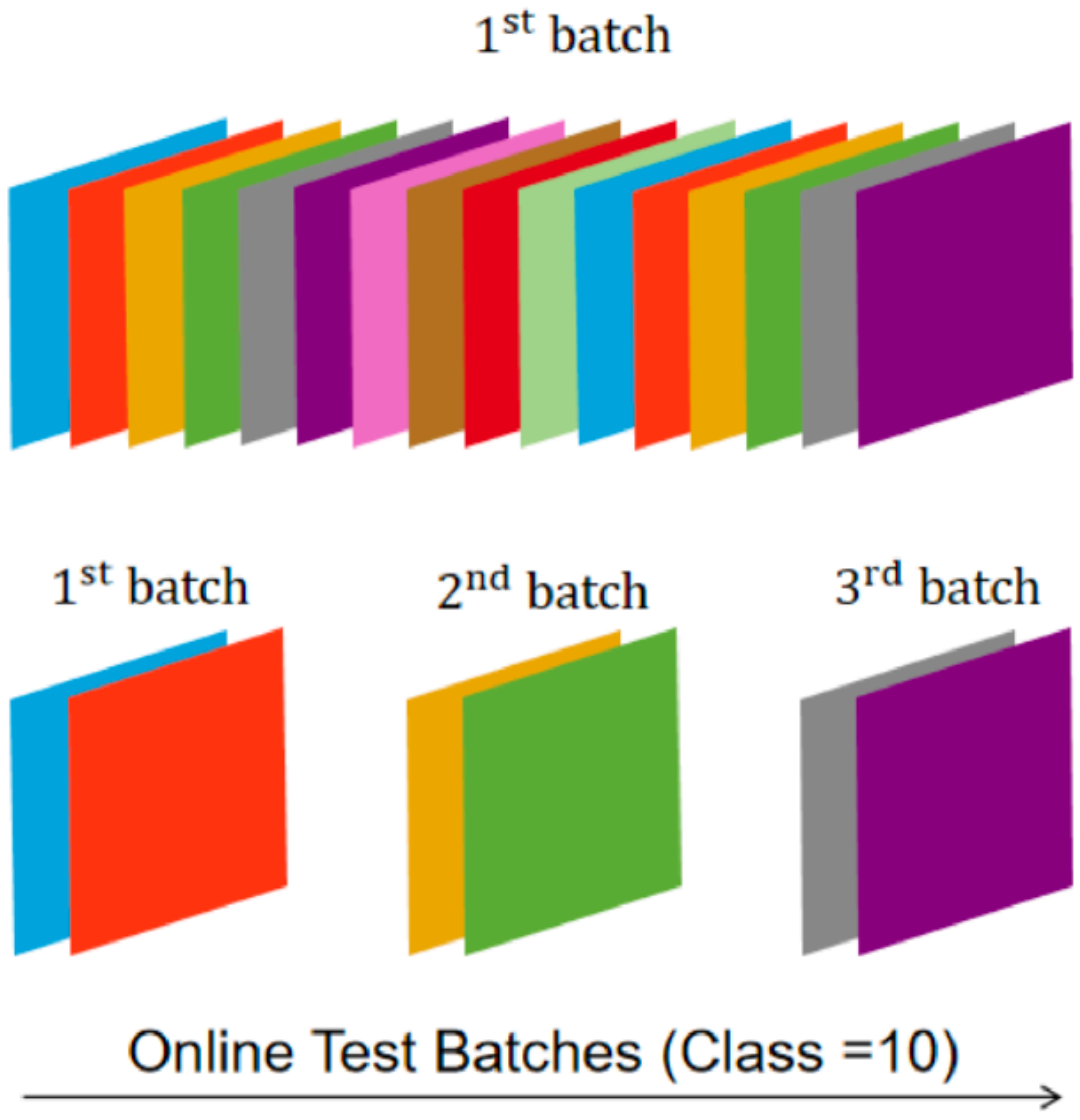
Normalization statistics are associated with domain characteristics.

Zixian Su, **Jingwei Guo**, et. al.Unraveling Batch Normalization for Realistic Test-Time Adaptation, *AAAI 2024*.

Proposed Solution #1:Test-time Exponential Moving Average

Mini-batch Model Degradation





Traditional TTA methods tend to degrade on small-batch data streams.

Zixian Su, **Jingwei Guo**, et. al.Unraveling Batch Normalization for Realistic Test-Time Adaptation, *AAAI 2024*.

Proposed Solution #1:Test-time Exponential Moving Average

Mini-batch Model Degradation

Proposition 1. *Given an infinite sample space where each sample is independently and identically distributed (i.i.d) with an equal probability of selection for each category. Let M denote the number of distinct categories contained within a given batch, and K be the category number in total. For a batch of size N, the expected number of unique categories (also referred to as category diversity) is given by:*

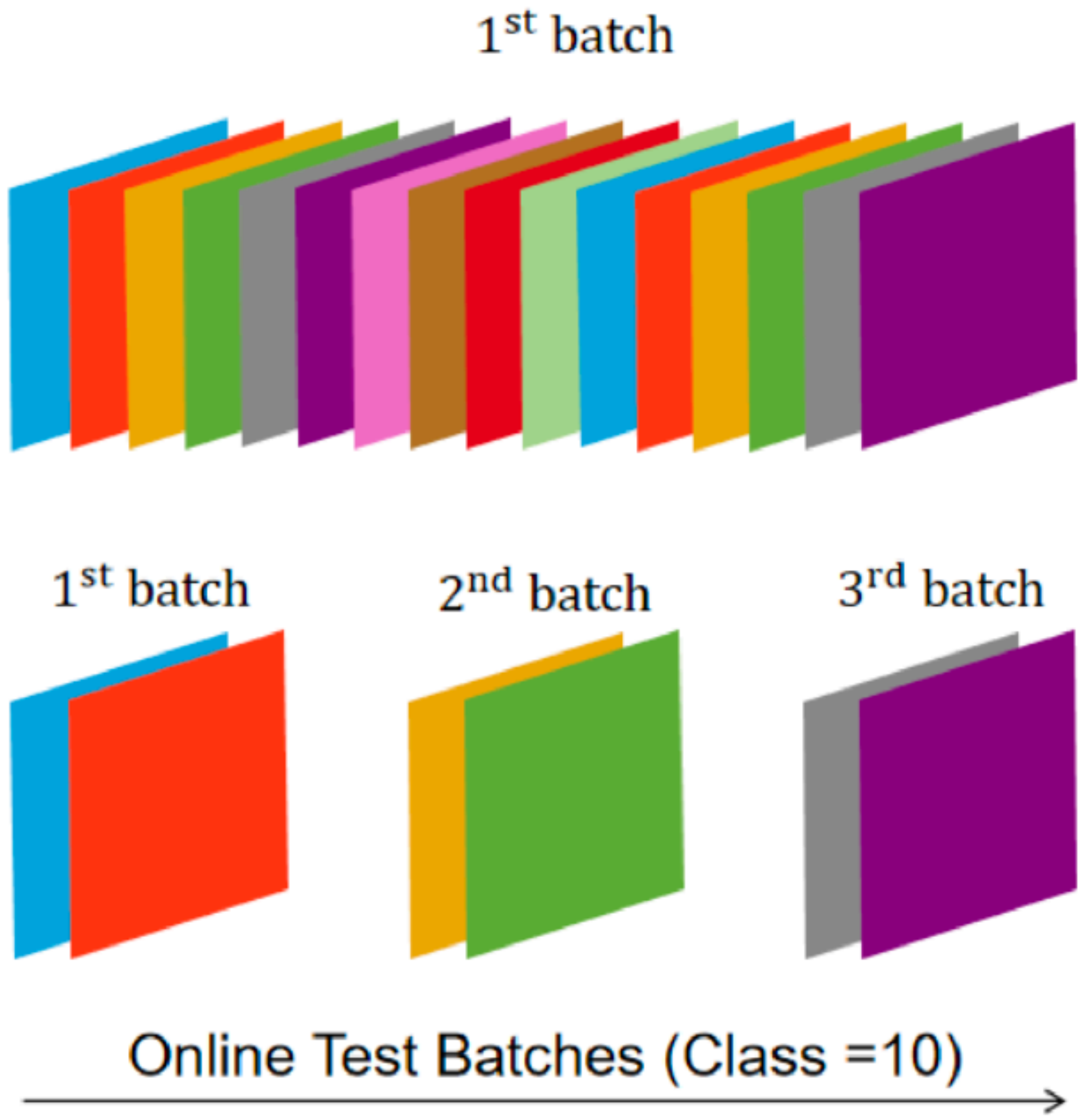
$$E(M|N) = \sum_{k=1}^{K} \left[k \cdot \frac{\mathbf{C}_{N-1}^{k-1} \mathbf{C}_{K}^{k}}{\mathbf{C}_{N+K-1}^{K-1}} \right], \quad (2)$$

where $\mathbf{C}$ *denotes the combination symbol in Combinatorics.*

Take CIFAR-10 dataset as an example:

$$N_t = 2, E(M_t|N_t) = 1.82 \ll E(M_s|N_s)$$

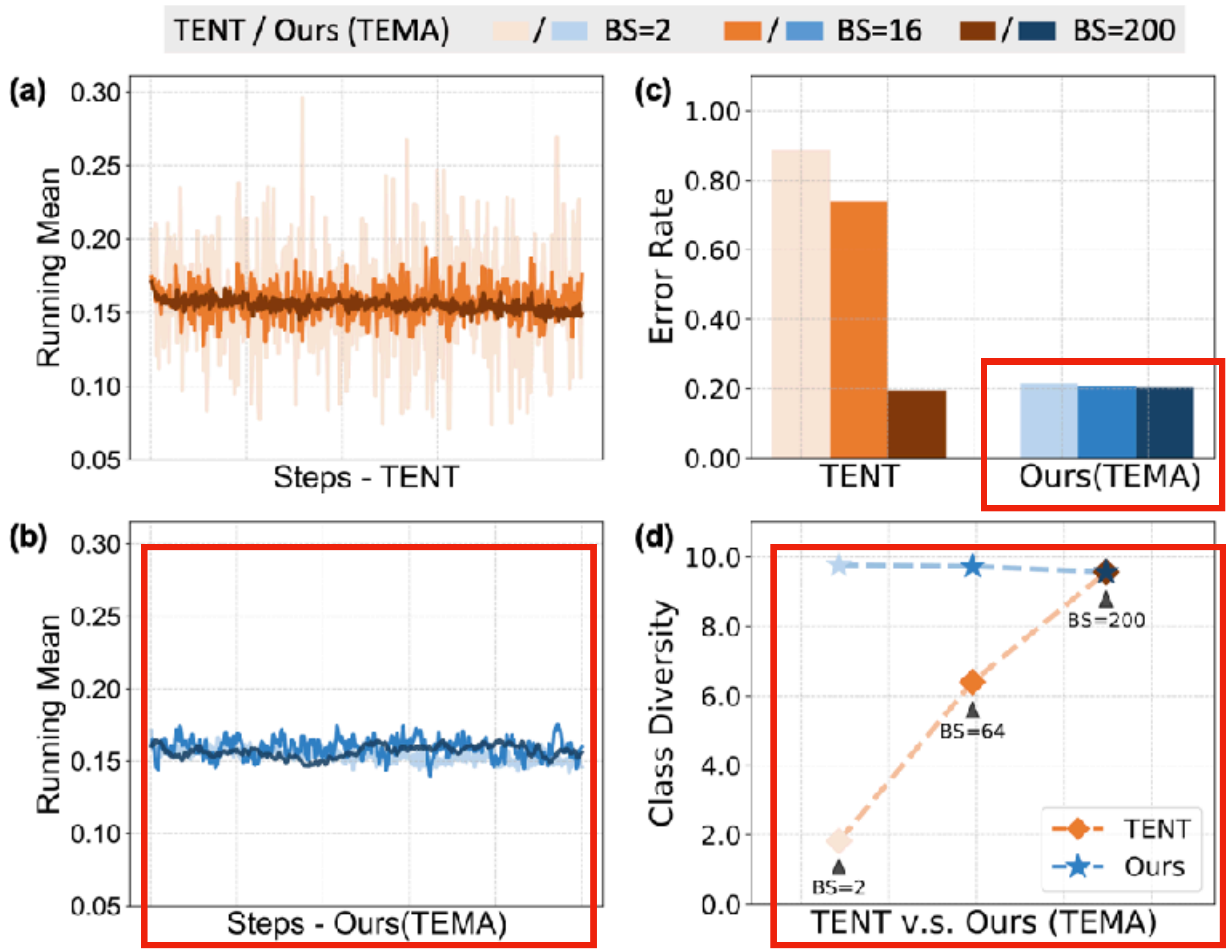
$$N_t = 200, E(M_t|N_t) = 9.57 \approx E(M_s|N_s)$$

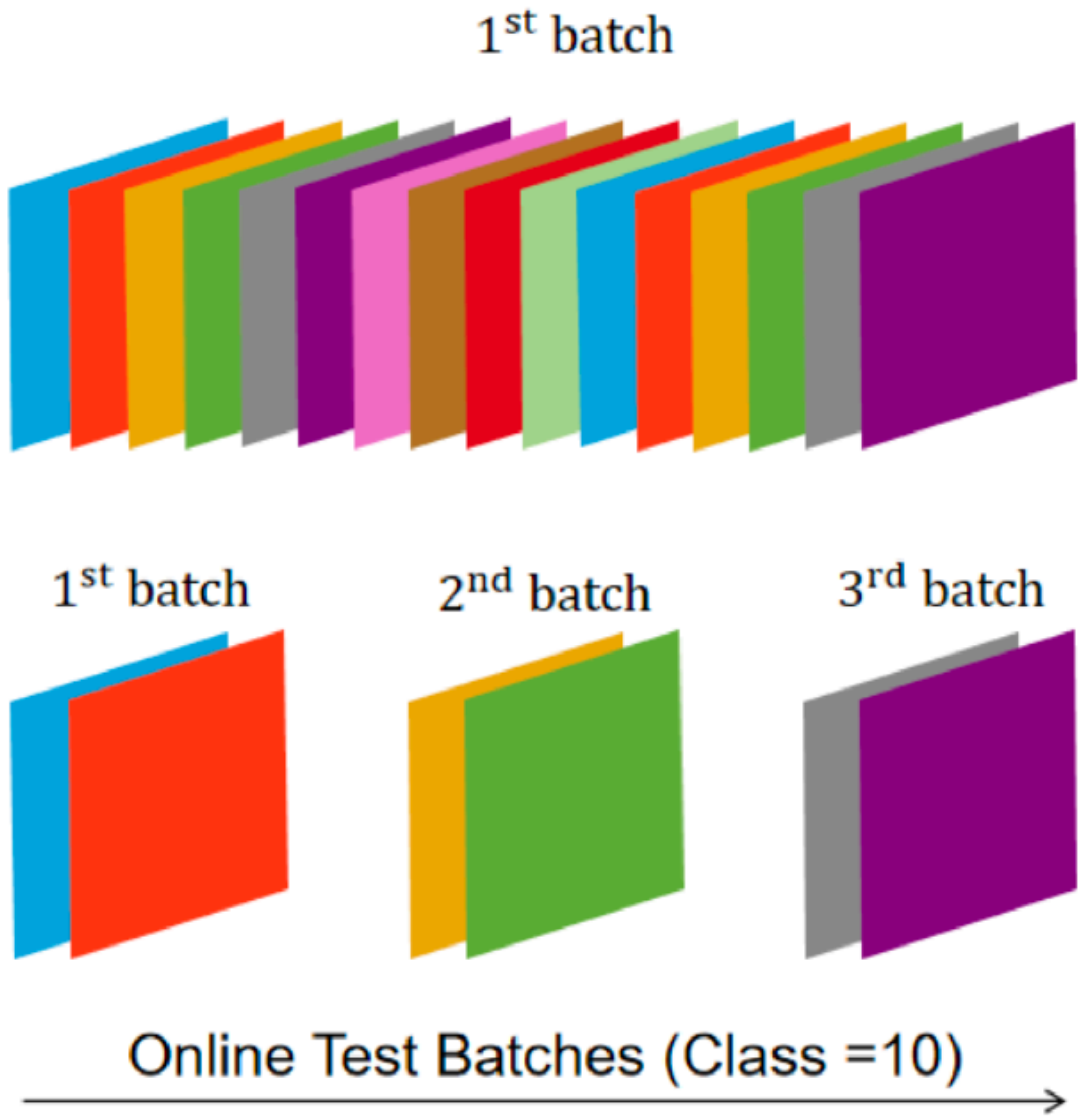


Class diversity discrepancy obscures true distribution shifts.

Zixian Su, **Jingwei Guo**, et. al.Unraveling Batch Normalization for Realistic Test-Time Adaptation, *AAAI 2024*.

Mini-batch Model Degradation





TTA methods degrade on small batches due to reduced class diversity.

Zixian Su, **Jingwei Guo**, et. al.Unraveling Batch Normalization for Realistic Test-Time Adaptation, *AAAI 2024*.

Proposed Solution #1:Test-time Exponential Moving Average

Exponential Moving Average at Test-time

$$\mu_t^{ema} = m \cdot \mu_t^{batch} + (1-m) \cdot \mu_{t-1}^{ema}$$

$$\sigma_t^{ema2} = m \cdot \sigma_t^{batch^2} + (1-m) \cdot \sigma_{t-1}^{ema2}$$

- Larger m emphasizes local batches
- Smaller m prioritize global statistics

Proposition 2. *Given the iterative rules of TEMA defined in Eq. 3 and 4, it yeilds the i-th term as a cumulative sum of the past batch statistics weighted by $w_0 = (1-m)^i$ for initial batch and $w_t = (1-m)^{(i-t)}m$ for $t = 1, 2, ..., i$. Let ϵ be a threshold defining the effective sample batch, such that only batches with a relative weight $w_t / w_i > \epsilon$ are included. Then, the expanded sample scope for statistical estimation in TEMA can be formally expressed as $\hat{N}_t = \lfloor \log_{1-m} \epsilon \rfloor \cdot N_t$.*

We apply grid search to tune m, guide by a tailored objective that balance local and global statistics.

$$\arg\min_m \left| \frac{E(M_s|N_s)}{E(M_t|N_t)} - 1 \right| + \lambda \cdot \frac{N_t}{N_s}$$

Aligning Class Diversity & Sample Scope

Zixian Su, **Jingwei Guo**, et. al.Unraveling Batch Normalization for Realistic Test-Time Adaptation, *AAAI 2024*.

Proposed Solution #1:Test-time Exponential Moving Average

Empirical Results — Error Reduction

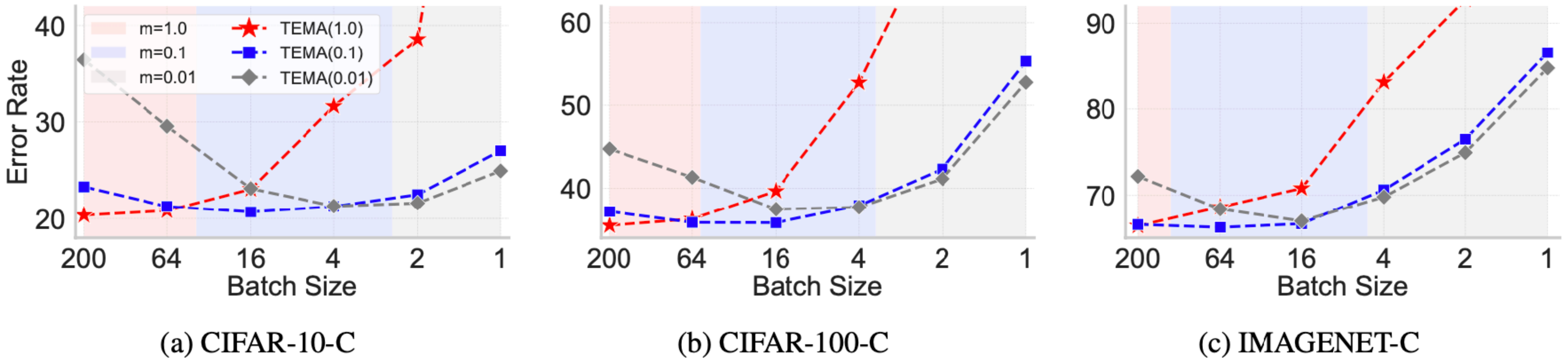
Continual		CIFAR-10-C							CIFAR-100-C						
		200	64	16	4	2	1	Avg.	200	64	16	4	2	1	Avg.
	Source	43.50	43.50	43.50	43.50	43.50	43.50	43.50	46.45	46.45	46.45	46.45	46.45	46.45	46.45
TR	TENT	19.55	26.32	74.07	85.07	88.69	90.00	63.95	61.12	86.49	96.07	98.40	98.79	99.00	89.98
TR	CoTTA	16.24	17.65	34.36	78.88	87.79	90.00	54.15	32.68	34.30	47.52	92.62	97.84	98.96	67.32
TR	SAR	20.40	20.74	22.89	31.35	40.32	89.83	37.59	31.90	35.89	54.84	66.08	73.24	98.91	60.14
TR	AdaCont.	18.50	17.41	19.69	35.01	63.81	31.55	31.00	33.61	35.40	55.04	89.45	96.16	62.67	62.06
TR	ETA	17.64	20.01	30.60	56.78	83.24	89.83	49.68	32.31	35.17	44.72	88.22	98.96	98.91	66.38
TF	TBN	20.35	20.82	23.06	31.62	38.57	89.83	37.38	35.50	36.29	39.67	52.73	73.24	98.91	56.06
TF	α-BN	30.60	30.67	30.89	31.89	32.91	34.47	31.91	37.02	38.27	35.92	37.17	37.25	41.18	37.80
TF	AdaptBN	20.36	20.71	21.98	26.79	32.19	37.52	26.59	35.40	**35.78**	35.92	37.14	39.23	41.85	37.55
TF	LAME	64.52	57.66	47.70	44.38	43.75	90.00	58.00	98.49	73.83	47.59	46.64	46.50	99.00	68.68
TF	**Ours**	**20.20**	**20.57**	**20.74**	**21.45**	**20.91**	**21.05**	**20.82**	**34.63**	36.11	**35.31**	**36.02**	**36.32**	**39.30**	**36.28**

Table 1: Continual adaptation on corruption benchmark CIFAR-10-C/CIFAR-100-C. Error rate (↓) averaged over 15 corruption with severity level 5 for each test batch size (200/64/16/4/2/1).

Zixian Su, **Jingwei Guo**, et. al.Unraveling Batch Normalization for Realistic Test-Time Adaptation, *AAAI 2024*.

Proposed Solution #1:Test-time Exponential Moving Average

Empirical Results — Momentum Analysis



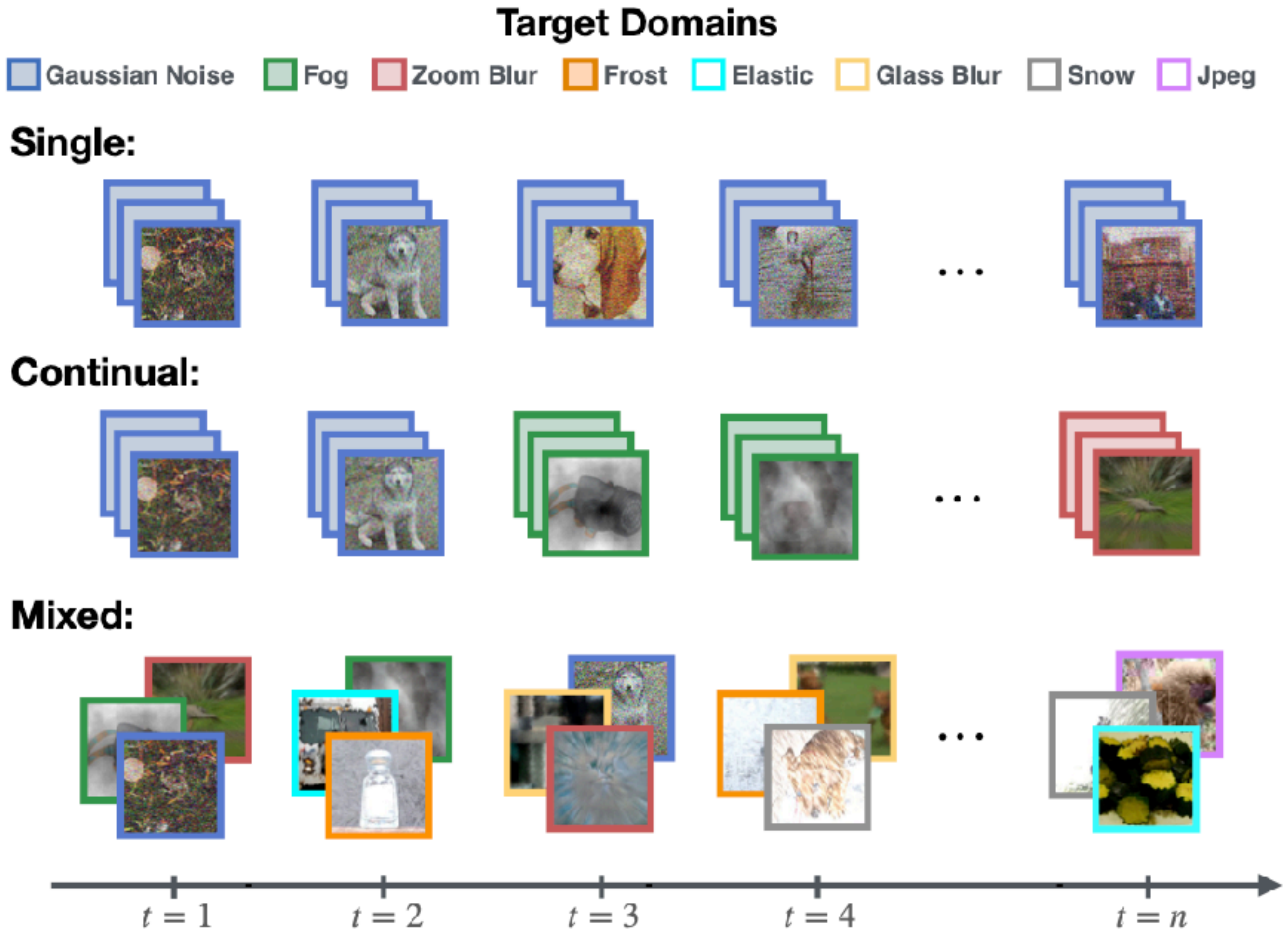
Our designed objective enables effective identification of the optimal momentum value that balances both local and global batch statistical information.

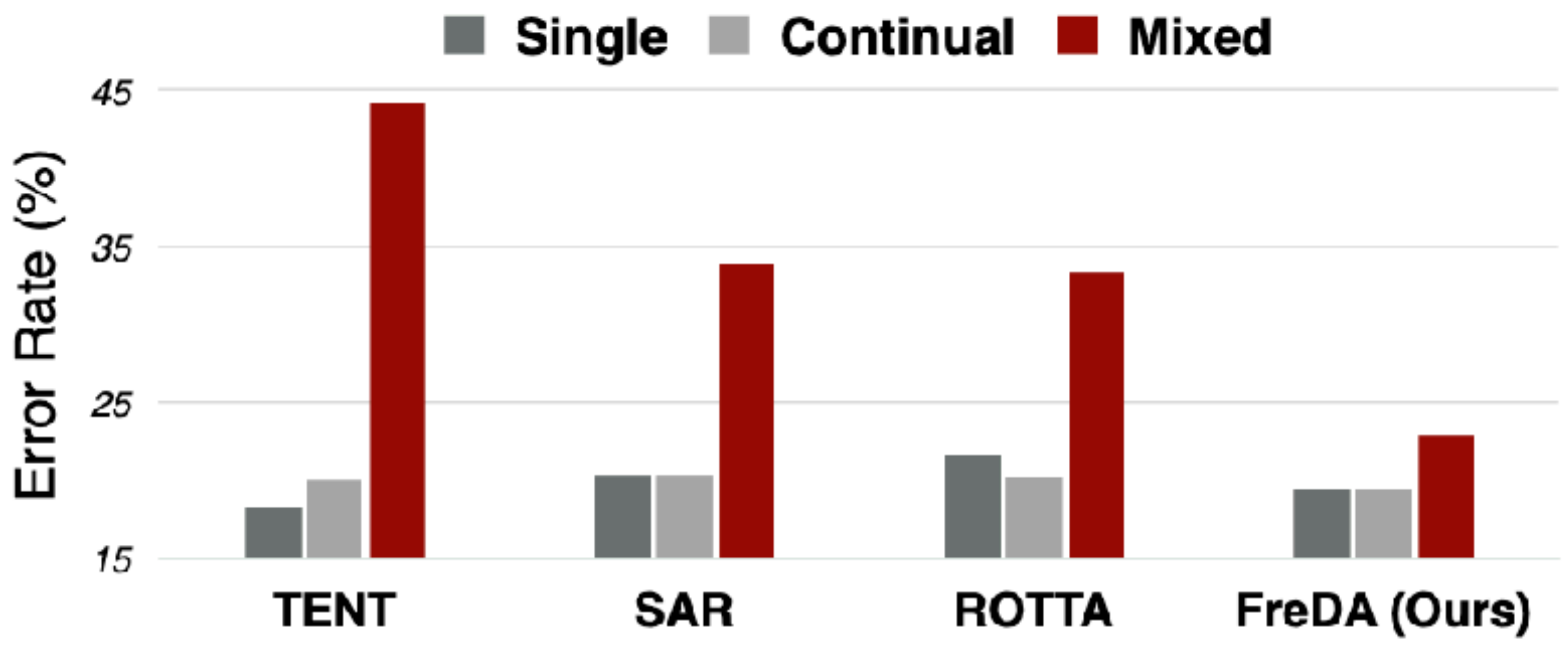
Zixian Su, **Jingwei Guo**, et. al.Unraveling Batch Normalization for Realistic Test-Time Adaptation, *AAAI 2024*.

Solution #2: Un-mixing TTA under Heterogeneous Data Streams

Proposed Solution #2: Frequency-based Decentralized Adapt.

Heterogeneous / Mixed-domain Model Degradation



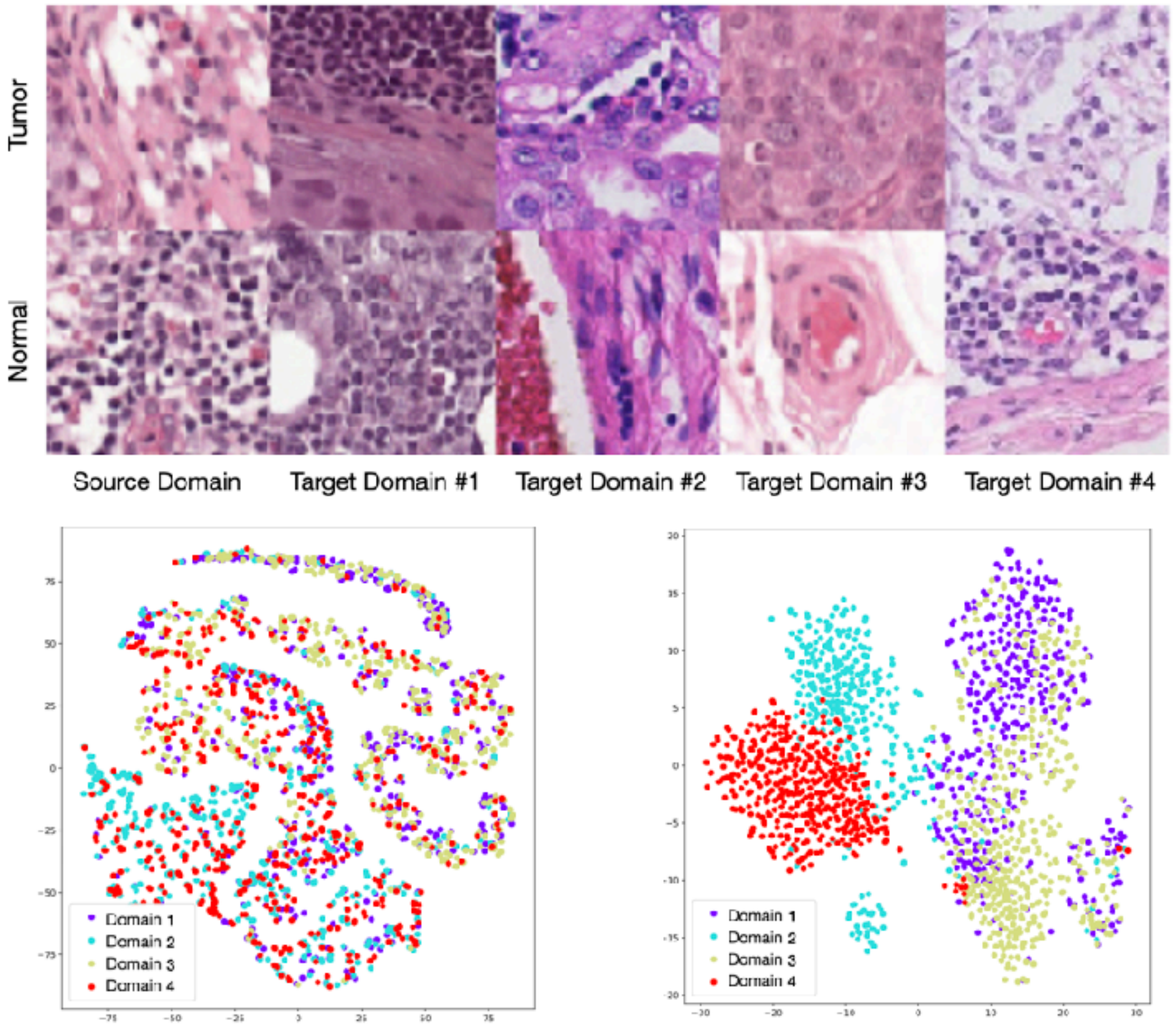


Conventional TTA methods assume homogenous data streams and employ whole-batch adaptation strategies.

Zixian Su*, **Jingwei Guo***, et. al., Un-mixing Test-time Adaptation under Heterogeneous Data Streams, *TKDE (Under Review) 2025*.

Proposed Solution #2: Frequency-based Decentralized Adapt.

Frequency-based Domain Separation

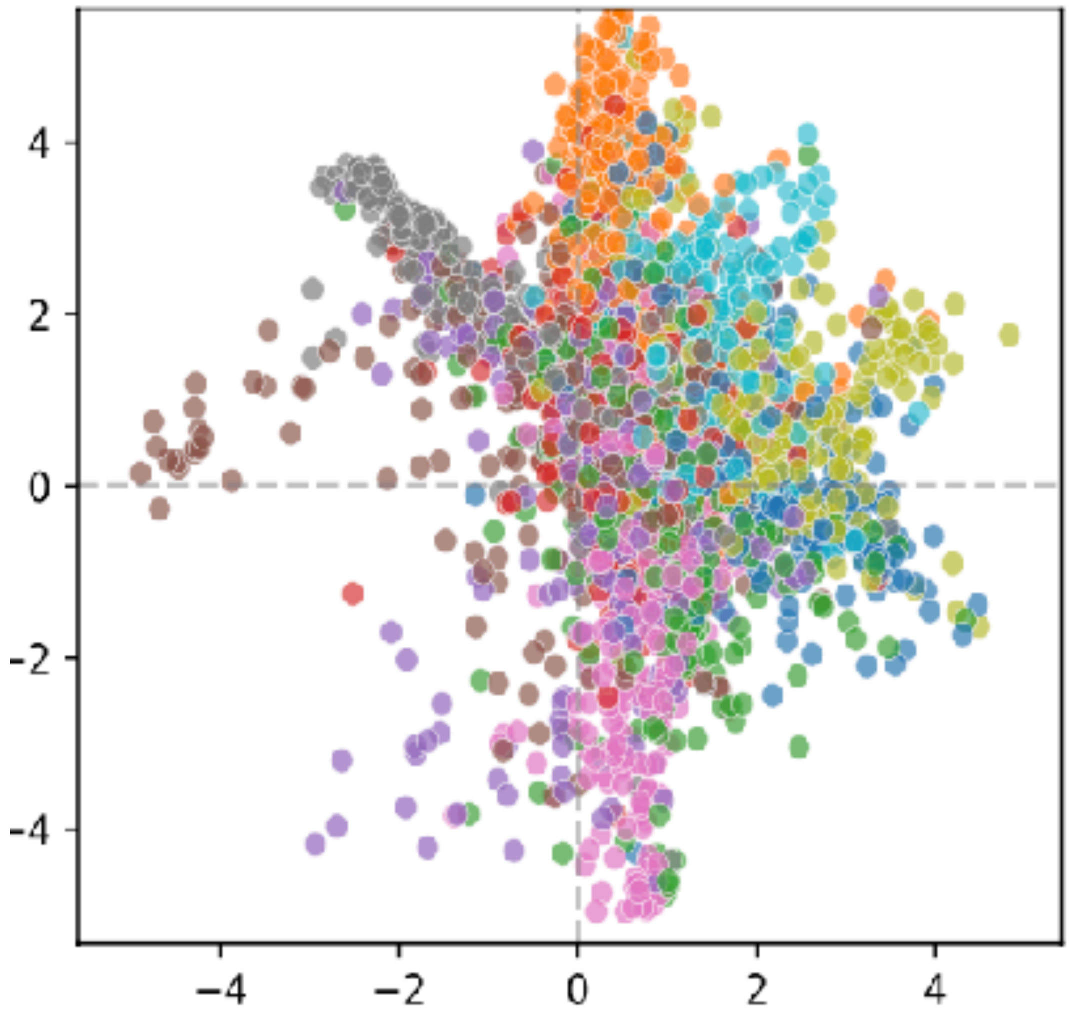


High-frequency information enable target subdomains' separation, contrasting with pre-trained model features.

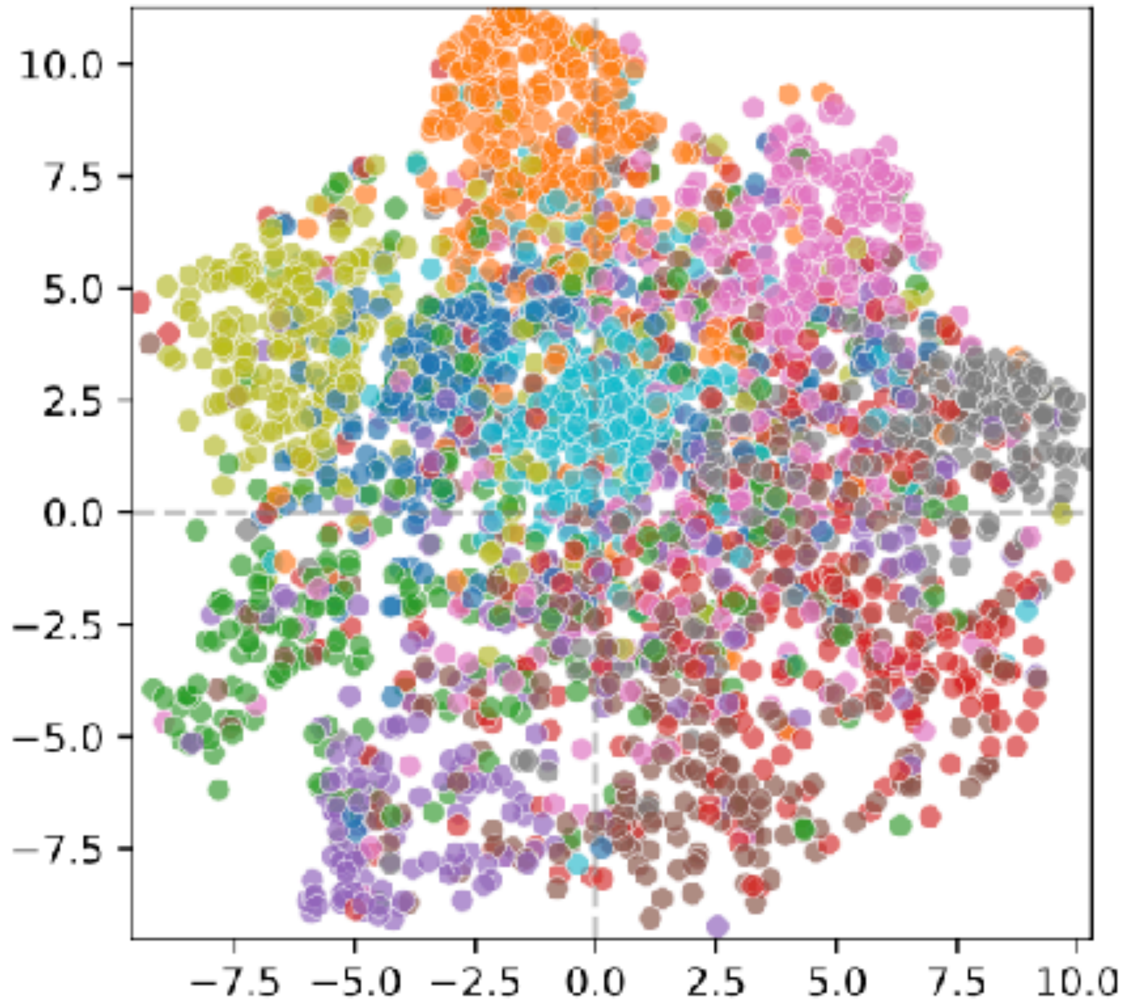
Error rate (%)

Methods	C10	C100	IN
TBN (Centralized)	33.8	45.8	82.5
TBN (Decentralized)	28.5	43.2	77.6

Class Separation



(a) Centralized Adaptation

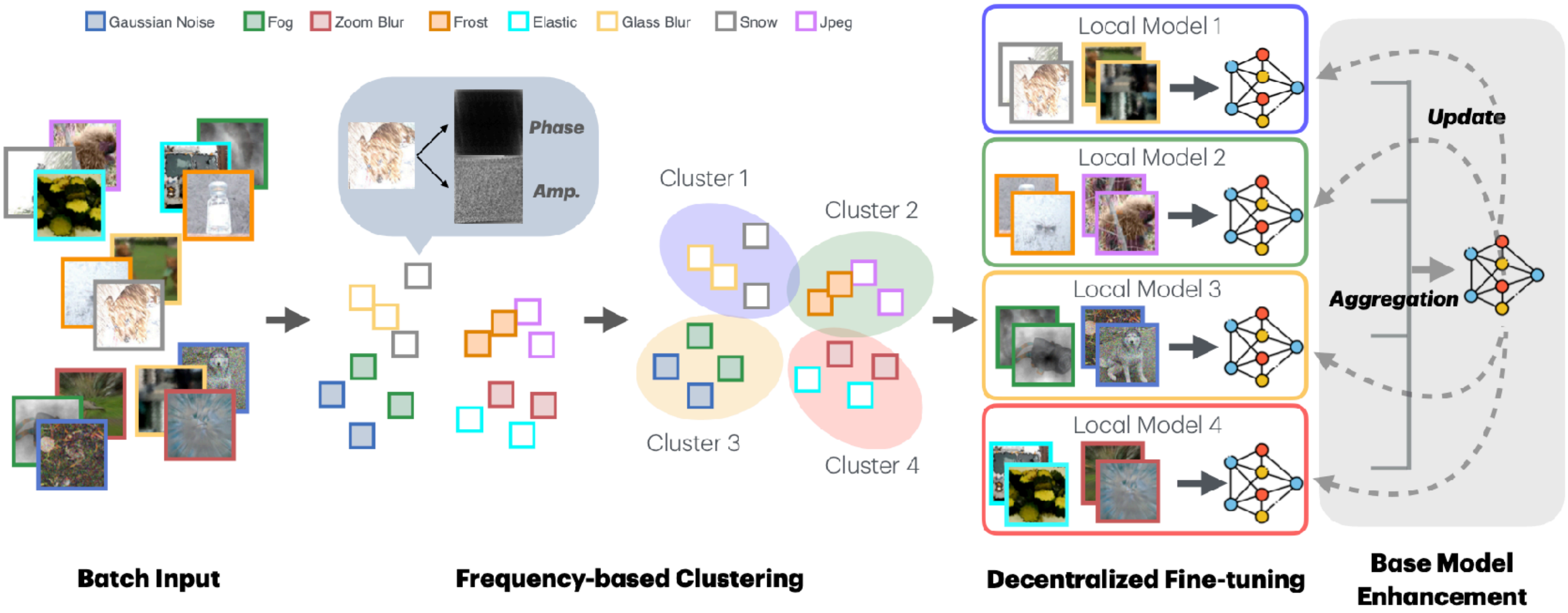


(b) Decentralized Adaptation

Zixian Su*, **Jingwei Guo***, et. al., Un-mixing Test-time Adaptation under Heterogeneous Data Streams, *TKDE (Under Review) 2025*.

Proposed Solution #2: Frequency-based Decentralized Adapt.

Our Overall Framework



Zixian Su*, **Jingwei Guo***, et. al., Un-mixing Test-time Adaptation under Heterogeneous Data Streams, *TKDE (Under Review) 2025*.

Proposed Solution #2: Frequency-based Decentralized Adapt.

Empirical Results

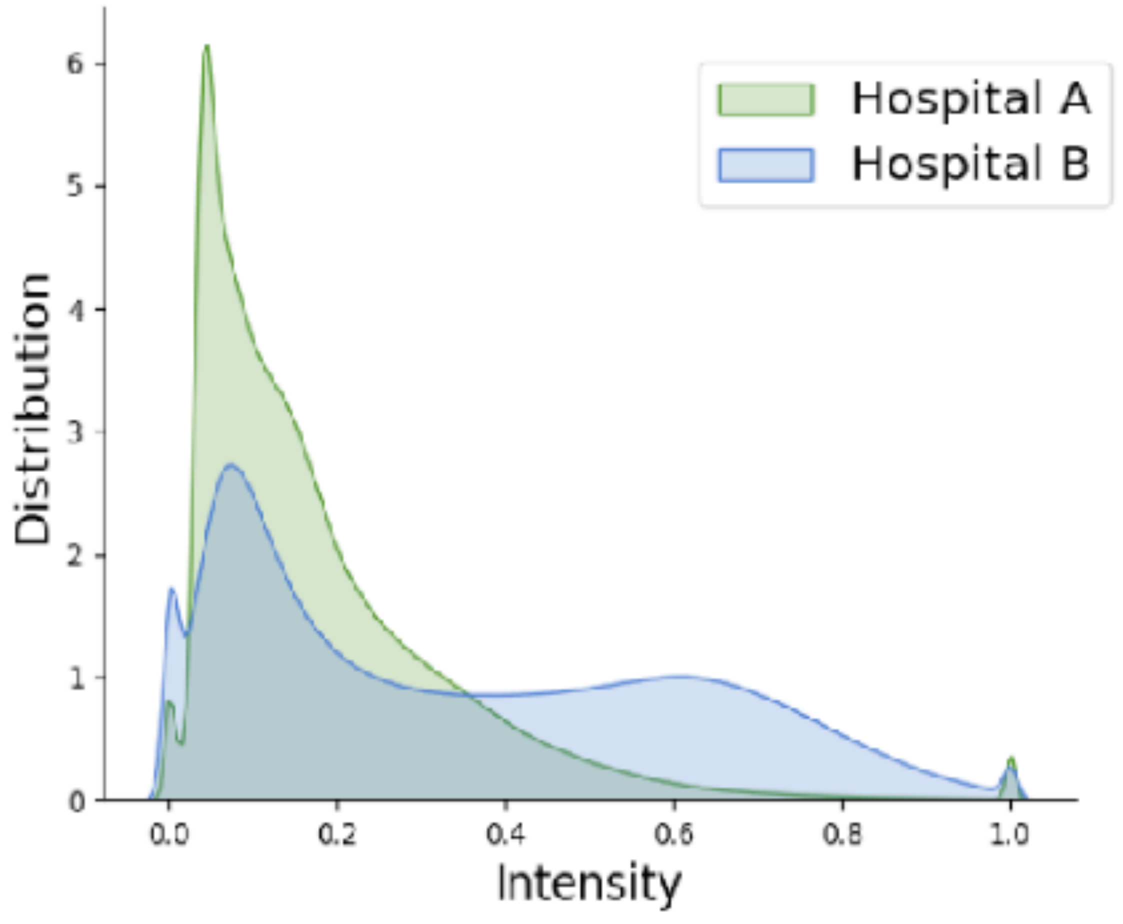
Baseline & Methods	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brig.	Contr.	Elast.	Pixel	JPEG	**Avg.**
CIFAR-10-C (WRN-28)	72.3	65.7	72.9	46.9	54.3	34.8	42.0	25.1	41.3	26.0	9.3	46.7	**26.6**	58.4	30.3	43.5
TBN	45.5	42.8	59.7	34.2	44.3	29.8	32.0	19.8	21.1	21.5	9.3	27.9	33.1	55.5	30.8	33.8
TENT (ICLR 21')	73.5	70.1	81.4	31.6	60.3	29.6	28.5	30.8	35.3	25.7	13.6	44.2	32.6	70.2	34.9	44.1
ETA (ICML 22')	36.2	33.3	52.3	22.9	38.9	22.4	20.5	19.5	19.7	20.4	11.3	35.4	**26.6**	38.8	25.1	28.2
AdaContrast (CVPR 22')	36.7	34.3	48.8	**18.2**	39.1	21.1	17.7	18.6	18.3	16.8	**9.0**	17.4	27.7	44.8	**24.9**	26.2
CoTTA (CVPR 22')	38.7	36.0	56.1	36.0	**36.8**	32.3	31.0	19.9	**17.6**	27.2	11.7	52.6	30.5	35.8	25.7	32.5
SAR (ICLR 23')	45.5	42.7	59.6	34.1	44.3	29.7	31.9	19.8	21.1	21.5	9.3	27.8	33.0	55.4	30.8	33.8
RoTTA (CVPR 23')	60.0	55.5	70.0	23.8	44.1	20.7	21.3	20.2	22.7	**16.0**	9.4	22.7	27.0	58.6	29.2	33.4
RDumb (NeurIPS 23')	34.9	32.3	49.4	23.3	38.2	23.3	20.7	19.9	19.3	20.7	11.2	29.3	26.7	41.5	25.2	27.7
DeYO (ICLR 24')	45.8	42.3	65.7	21.3	41.8	25.1	19.5	21.1	19.6	19.2	12.3	21.8	28.5	39.3	28.0	30.1
UnMix-TNS (ICLR 24')	50.0	44.4	44.3	34.4	48.2	32.7	30.0	35.5	35.9	47.5	28.1	38.7	43.9	40.0	43.3	39.8
FreDA (ours)	**23.1**	**22.2**	**32.2**	18.7	41.6	**18.8**	**16.8**	**17.9**	19.9	16.9	9.8	**13.2**	29.1	**35.4**	28.6	**22.9**
CIFAR-100-C (ResNeXt-29)	73.0	68.0	39.4	29.3	54.1	30.8	28.8	39.5	45.8	50.3	29.5	55.1	37.2	74.7	41.2	46.4
TBN	62.7	60.7	43.1	35.5	50.3	35.7	34.4	39.9	51.5	**27.5**	45.5	42.3	72.8	46.4	45.8	45.8
TENT (ICLR 21')	95.6	95.2	89.2	72.8	82.9	74.4	72.3	78.0	79.7	84.7	71.0	88.5	77.8	96.8	78.7	82.5
ETA (ICML 22')	42.6	40.3	**34.1**	30.3	42.4	32.0	29.4	35.6	35.8	44.1	30.2	41.8	**36.9**	38.9	40.9	37.0
AdaContrast (CVPR 22')	54.5	51.5	37.6	30.7	45.4	32.1	30.3	36.9	36.5	45.3	28.0	42.7	38.2	75.4	41.7	41.8
CoTTA (CVPR 22')	54.4	52.7	49.8	36.0	45.8	36.7	33.9	38.9	35.8	52.0	30.4	60.9	40.2	38.0	41.1	43.1
SAR (ICLR 23')	75.8	72.7	41.1	**29.2**	45.2	31.1	28.9	36.7	37.7	43.9	29.3	41.8	37.1	89.2	42.4	45.5
RoTTA (CVPR 23')	65.0	62.3	39.3	33.4	50.0	34.2	32.6	36.6	36.5	45.0	**26.4**	41.6	40.6	89.5	48.5	45.4
RDumb (NeurIPS 23')	42.3	40.0	**34.1**	30.5	42.4	31.9	29.5	35.7	35.9	43.6	30.4	41.9	36.9	38.1	40.5	36.9
DeYO (ICLR 24')	57.2	53.4	38.8	34.7	47.3	37.3	34.1	40.8	40.5	50.6	33.3	45.8	41.5	94.5	45.7	46.4
UnMix-TNS (ICLR 24')	65.8	64.1	46.4	37.5	51.7	36.0	36.4	38.5	39.4	51.1	29.3	42.8	43.2	67.8	49.4	46.6
FreDA (ours)	**34.8**	**34.7**	36.6	29.4	**41.2**	**29.9**	**28.4**	**33.8**	**33.7**	41.1	29.8	**34.9**	**36.9**	**37.1**	**38.7**	**34.7**
IN-C (ResNet-50)	97.8	97.1	98.2	81.7	89.8	85.2	77.9	83.5	77.1	75.9	41.3	94.5	82.5	79.3	68.6	82.0
TBN	92.8	91.1	92.5	87.8	90.2	87.2	82.2	82.2	82.0	79.8	48.0	92.5	83.5	75.6	70.4	82.5
TENT (ICLR 21')	99.2	98.7	99.0	90.5	95.1	90.5	84.6	86.6	84.0	86.5	46.7	98.1	86.1	77.7	72.9	86.4
ETA (ICML 22')	90.7	89.2	90.5	77.0	**80.6**	74.0	68.9	72.4	70.3	64.6	43.9	93.4	**69.2**	**52.3**	**55.9**	72.9
AdaContrast (CVPR 22')	96.2	95.5	96.2	93.2	96.4	96.3	90.5	92.7	91.9	92.4	50.8	97.0	96.6	89.7	87.1	90.8
CoTTA (CVPR 22')	89.1	86.6	88.5	80.9	87.2	81.1	75.8	73.3	75.2	70.5	41.6	85.0	78.1	65.6	61.6	76.0
SAR (ICLR 23')	98.4	97.3	98.0	84.0	87.3	82.6	77.2	77.5	76.1	72.5	43.1	96.0	78.3	61.8	60.4	79.4
RoTTA (CVPR 23')	89.4	88.6	89.3	83.4	89.1	86.2	80.0	78.9	76.9	74.2	**37.4**	89.6	79.5	69.0	59.6	78.1
RDumb (NeurIPS 23')	89.0	87.6	88.6	78.1	82.3	75.2	70.1	73.0	71.0	65.1	43.9	92.6	70.7	53.7	56.3	73.1
DeYO (ICLR 24')	99.5	99.2	99.5	89.5	95.0	83.9	78.8	75.0	87.8	79.2	47.3	99.2	92.4	59.0	60.4	83.0
UnMix-TNS (ICLR 24')	91.7	92.8	91.7	92.3	93.4	91.5	84.8	86.3	84.1	85.0	62.0	96.5	88.6	81.7	77.3	86.7
FreDA (ours)	**72.4**	**74.0**	**71.4**	**76.5**	82.3	**72.1**	**64.1**	**64.4**	**64.8**	**59.1**	43.7	**79.7**	71.0	54.2	58.6	**67.2**

Error Rate -3.73% under Mixed Distribution Shifts

Zixian Su*, **Jingwei Guo***, et. al., Un-mixing Test-time Adaptation under Heterogeneous Data Streams, *TKDE (Under Review) 2025*.

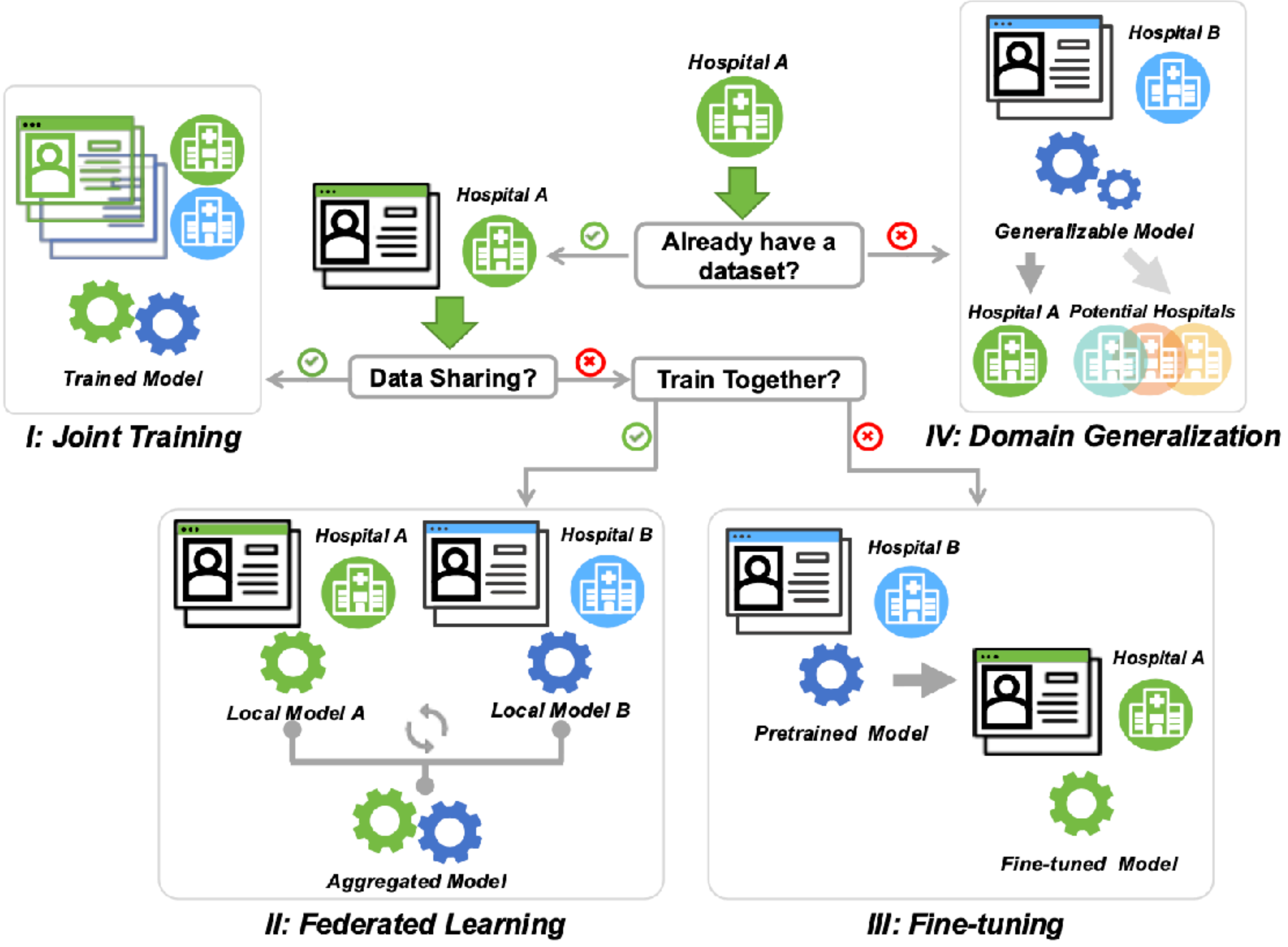
Solution #3: Navigating Distribution Shifts in Medical Image Analysis: A Survey

Proposed Solution #3: Navigating Distribution Shifts in MedIA



Suppose Hospital A (target) seeks a model tailored for its distribution, developed in collaboration with Hospital B (source):

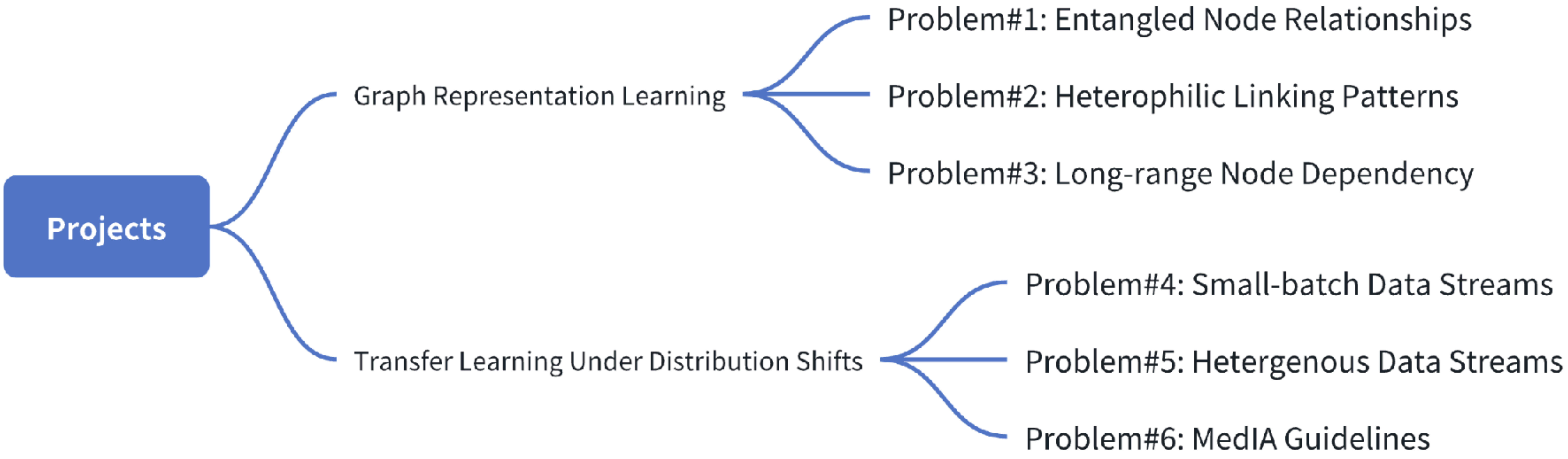
- Data Accessibility
- Privacy Concerns
- Collaborative Protocols



Zixian Su*, **Jingwei Guo***, et.al. Navigating Distribution Shifts in Medical Image Analysis. *ACM Comput. Surv (Under Review) 2025.*

Research Summary

Towards Artificial Intelligence for Realistic World Problems



Thanks