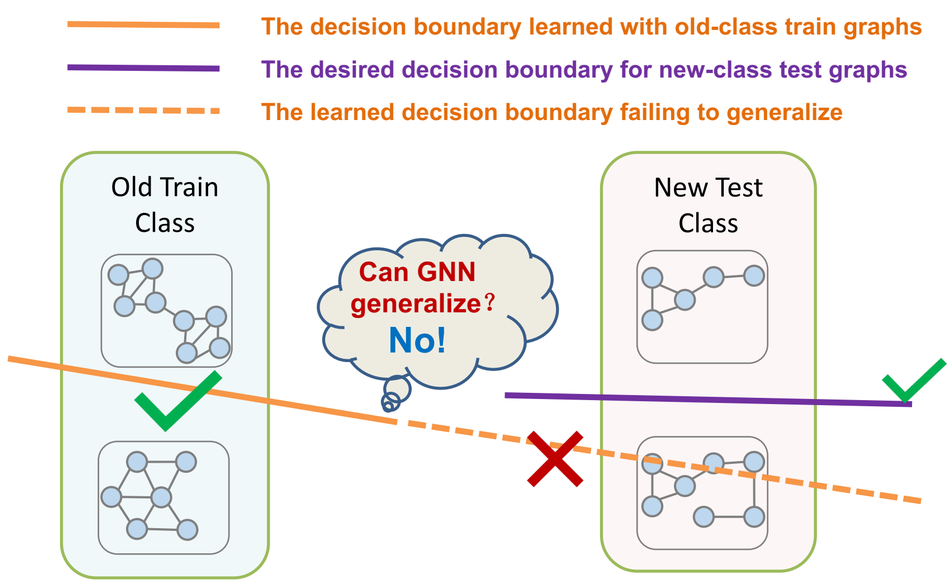
GLNCD: Graph-Level Novel Category Discovery

B. Deng, L. Fu, S. Huang, T. Liao, J. Chen, T. Zhang, C. Chen

New Problem, Benchmark, and Baselines



- **New Problem:**
  - Graph-level classification assumes all categories/classes known during training, **fails** to handle **test-time new classes in *real-world* scenarios**
  - **GLNCD: Classification for old-classes + clustering for new-classes**

GLNCD Dataset	# Graphs	Avg. # nodes	Avg. # edges	# node/edge feats.
ENZYMES	600	32.6	124.3	21/0
MalNet-Tiny	5000	1410.3	2859.9	0/0
REDDIT12K	11929	391.41	456.89	0/0
CIFAR10	60000	117.6	941.1	5/1

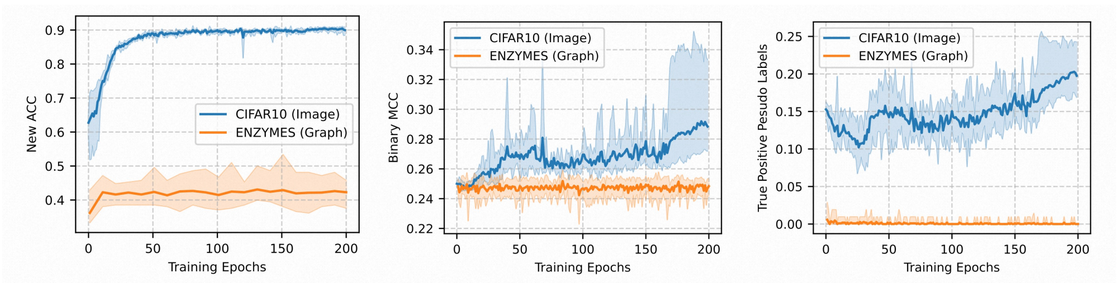
- **New Benchmark:** Graph-level NCD datasets built with 4 multi-class graph datasets
  - **Four domains:** Bioinformatics, Program Analysis, Social Networks, and Computer Vision
  - **Diverse sizes:** From small to large dataset size
  - **Evaluation Protocol:** Old/new class split and evaluation metric
- **New Baselines :** Visual NCD methods AutoNovel, NCL, and DualRS are adapted to graph data by replacing the backbone and self-supervised learning (SSL) with
  - **GNN Backbone:** Simple and strong graph-level GNNs from [1]
  - **Graph SSL:** Pervasive graph-level representation learning method GraphCL [2]

Challenges in NCD Method Adaptation: From Image to Graph Data

Dataset	Image datasets			Graph datasets			
	CIFAR10	CIFAR 100	SVHN	ENZYMES	MalNet-Tiny	REDDIT12K	CIFAR10 (Graph)
Old ACC (Test)	95.34	74.51	98.10	73.00	93.30	67.59	61.36
New ACC (Train)	88.50	74.28	94.21	41.90	74.51	39.21	41.67
Gap: Old (row1) – New (row2	**6.84**	**0.23**	**3.89**	**31.10**	**18.79**	**28.38**	**19.69**

Table 2: The average performance (over 10 runs) of AutoNovel [3] on image and graph datasets (The AutoNovel on graph datasets are the adapted version**). Old ACC (Test)** is the accuracy on the old-class samples in test dataset. **New ACC (Train)** is the clustering accuracy on the unlabeled training dataset. The last row is the gap between these metrics.

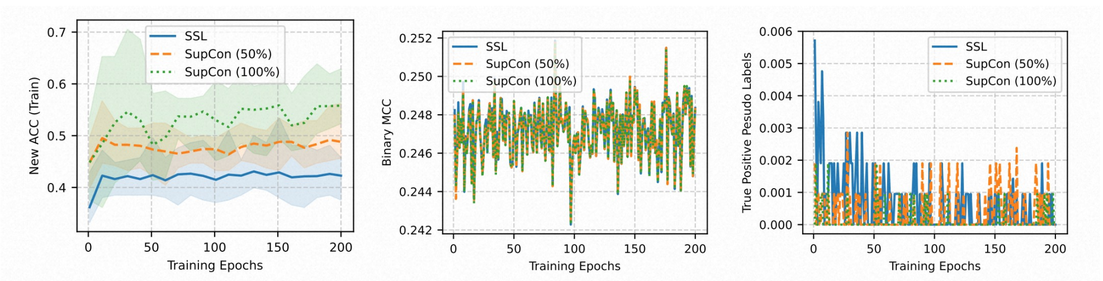
- **Direct Adaptation Fails:** The methods adapted from image domain fail on graphs
  - **The extremely large new–old performance gap** (last row of **Table 2**) observed on graph datasets, compared with image datasets, suggests that simply adapting visual NCD methods designed for images is inadequate



(a) The clustering accuracy on unlabeled new-class samples. (b) The MCC (quality) of RS pseudo labels for sample pairs. (c) The ratio of true positive RS pseudo labels for sample pairs.

Figure 1: Training dynamics of AutoNovel [3] on CIFAR10 (image) and ENZYMES (graph). **(a)** The performance on unlabeled training dataset. **(b)** The Matthews Correlation Coefficient (MCC) to evaluate the quality of pairwise pseudo-labels generated via ranking statistics (RS). **(c)** The ratio of samples for which at least one true positive (same-class) pair is identified by RS. The definitions of these two pseudo-label metrics and the rationale for their selection are provided in Appendix A.

- **Why Direct Adaptation Fails? Ranking Statistics (RS) Fails:** NCD methods use RS to generate pseudo pairwise labels for new-class samples
  - **RS fails on Graphs:** As shown in Figure 1b and 1c, RS produces higher-quality pseudo-labels on CIFAR-10 (image) but lower-quality ones on ENZYMES



(a) The clustering accuracy on unlabeled new-category samples. (b) The MCC (quality) of RS pseudo labels for sample pairs. (c) The raito of true positive RS pseudo labels for sample pairs.

Figure 2: Training dynamics of G-AutoNovel [3] on ENZYMES (graph). Three GIN encoders are pretrained with 0% (SSL), 50% (SupCon), and 100% (SupCon), true binary pairwise labels. **(a)** The performance on unlabeled training dataset. **(b)** The Matthews Correlation Coefficient (MCC) to evaluate the quality of RS pseudo-labels. **(c)** The ratio of samples for which at least one true positive pair is identified by RS. The details about of these pseudo-label metrics are provided in Appendix A.

- **Why RS Fails? Insufficient Exploration of Graph Structure:** We pretrain three GNN encoders using SupCon with 0%, 50%, and 100% ground-truth pairwise labels to induce increasing representation quality, then compare RS pseudo-label quality and NCD performance across these settings.
  - **High-quality graph representation helps GLNCD but not via improving RS:** As shown in **Figure 2**, more oracle pairwise labels leads to better NCD (2a), but the pseudo-label quality (2b and 2c) does not improve accordingly.
  - **Hypothesize 1:** RS's lack of graph-structural information prevents it from improving pseudo-label quality.
  - **Improve Direction:** Structure-aware representations + structure-aware RS

Proposed Method: ProtoFGW-NCD

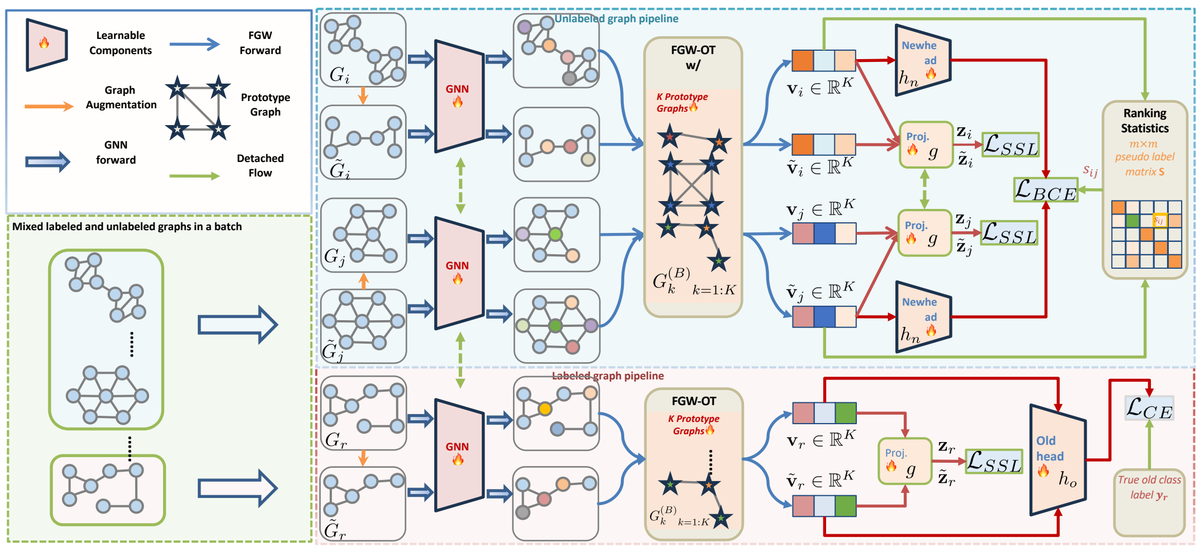


Figure 3: Illustration of our ProtoFGW-NCD

- **ProtoFGW-NCD:** utilizes **Fused Gromov-Wasserstein** (**FGW**) optimal transport to exploit structural information
  - ProtoFGW-CL: Structure-aware self-supervised learning of **ProtoFGW-NCD**
  - FGW-RS: Structure-aware RS for pseudo-labeling of **ProtoFGW-NCD**
  - Learnable Prototypes in **ProtoFGW-NCD**: enable efficient cross-view graph comparisons through prototype alignment
  - Unified Training: Representation learning + NCD training in one stage
  - Validate **Hypothesize 1:** we compare **ProtoFGW-NCD** and AutoNovel, identical except for the Bregman Alternating Projected Gradient (BAPG) layer that injects structural information via FGW; if ProtoFGW-NCD performs better, **Hypothesize 1** is supported.

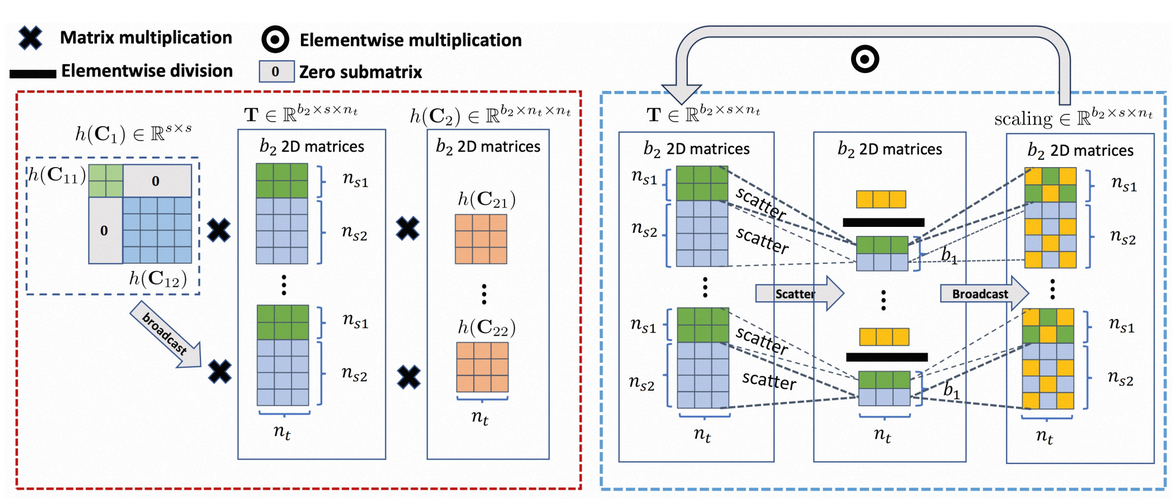


Figure 5: The illustration of two key operators in the forward of our BAPG layer, which parallels POT [4] BAPG solver on GPU. See Appendix B for the details.

- **BAPG Layer:** Differentiable FGW computation between two batches of sparse graphs of any sizes on GPU
  - Parallelizes operations for comparing a batch of graphs and their augmentations
  - Significant Speedup: **up to 2070×** faster than POT [4] BAPG implementation

Experiments

Dataset	ENZYMES		MalNet-Tiny		REDDIT12K		CIFAR10		Avg. Rank		
Method	**Old ACC**	**New ACC**	**Old ACC**	**New ACC**	**Old ACC**	**New ACC**	**Old ACC**	**New ACC**	**Old ACC**	**New ACC**	**ALL**
K-means	39.67± 1.83	38.86± 2.89	66.60± 9.26	58.72± 2.76	42.34± 4.28	37.05± 2.11	42.27± 0.12	40.72± 1.42	5.00	3.75	4.375
AutoNovel	71.33± 2.74	41.52± 1.86	80.30± 6.79	62.43± 5.40	68.91± 0.33	39.08± 1.34	61.10± 3.12	41.26± 1.31	3.00	2.00	2.500
NCL	67.67± 1.49	39.71± 3.77	85.50± 2.35	62.23± 1.68	69.35± 1.11	37.01± 1.20	70.63± 0.46	39.64± 0.82	2.25	3.50	2.875
DualRS	64.67± 3.21	39.33± 5.07	68.75± 7.45	49.53± 3.74	66.47± 0.87	40.76± 2.77	70.90± 0.86	39.17± 0.86	3.50	3.75	3.625
Ours	72.17± 5.67	44.84± 3.07	80.95± 6.16	63.35± 1.19	69.43± 3.74	40.81± 2.16	71.25± 0.61	38.92± 0.49	**1.25**	**2.00**	**1.625**

Table 4: The GLNCD results of various methods with **GCN+** backbone

Better performance: ProtoFGW-NCD achieves the best avg. rank on 4 datasets

Batch Size	64	64	64	128	128	128	256	256	256	512	512	512
Dataset	**POT**	**Ours**	**↑**	**POT**	**Ours**	**↑**	**POT**	**Ours**	**↑**	**POT**	**Ours**	**↑**
CSBM-20-10	8.90	0.04	250.7	17.96	0.02	719.6	34.19	0.03	1296.8	67.34	0.03	**2070.2**
CSBM-50-10	8.69	0.03	333.5	16.67	0.03	598.8	32.57	0.04	844.8	65.02	0.06	1020.1
CSBM-100-10	8.55	0.03	289.3	16.78	0.04	416.9	32.54	0.07	457.6	65.14	0.14	474.6

Table 6: The average batch time (s) of different BAPG implementations. ↑ indicates speedup factor

Impressive speedup: Our BAPG layer solves batch-to-batch **FGW** problems efficiently

Reference

[1] Luo, Yuankai, Lei Shi, and Xiao-Ming Wu. "Can Classic GNNs Be Strong Baselines for Graph-Level Tasks? Simple Architectures Meet Excellence." In: ICML 2025.

[2] You, Yuning, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. "Graph Contrastive Learning with Augmentations." In: NeurIPS 2020.

[3] Han, Kai, Sylvestre-Alvise Rebuffi, Sébastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. "AutoNovel: Automatically Discovering and Learning Novel Visual Categories." In: IEEE TPAMI.

[4] Flamary, Rémi, Nicolas Courty, Alexandre Gramfort, et al. "POT: Python Optimal Transport." In: JMLR.