A Systematic Comparison of Regularization Strategies for Few-Shot Learning on Graphs

Nicola Maestri
University of Trento
Trento, Italy

Abstract

Few-Shot Node Classification (FSNC) and Few-Shot Graph Classification (FSGC) are fundamental problems in Few-Shot Learning on Graphs, the research area that combines Graph Representation Learning and Few-Shot Learning (FSL). Despite extensive work in both domains, comprehensive studies regarding the integration of regularization and FSL strategies remain rare. In this paper, we establish baselines for more advanced methods by systematically evaluating how well models can generalize from few samples using five widely-adopted strategies: regularization, augmentation, prototypical networks, siamese networks, pre-training and fine-tuning. For each task, the same model is trained from scratch with an increasing number of samples per class. To leverage contextual information, the architecture consists of a sequence of three graph convolutional layers, which provides good inductive bias despite limited training data. Six datasets are considered for FSNC and three for FSGC, reporting as evaluation metrics micro accuracy and micro F1-score, averaged over multiple runs. Although the final performance correlates with training sample size and quality, the contribution of each strategy can be analyzed by focusing on the relative gain compared to the other methods. Results on the Mutag dataset show that Prototypical Networks excel with simple datasets having clearly separable classes, making the model more robust to outliers. Graph rewiring leads to improvements on citation graphs, suggesting our proposed heuristic benefits the task. In contrast, Cosine regularization yields the greatest performance improvements on coauthorship graphs and the Amazon Computers dataset. Overall, these preliminary findings represent a solid baseline for more advanced models which adapt prior knowledge to the presented tasks.

CCS Concepts

• **To be defined**;

Keywords

To be defined

ADV ML Project, May 2025, Treto, Italy

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Nicola Maestri. 2025. A Systematic Comparison of Regularization Strategies for Few-Shot Learning on Graphs. In . ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Despite advances in both graph neural networks and few-shot learning, their combination remains understudied. Few-Shot Node Classification (FSNC) and Few-Shot Graph Classification (FSGC) present unique challenges: models must generalize from extremely limited labeled data while capturing complex topological relationships in domain-specific graph structures, from molecular graphs to citation networks. In this work, we address this methodological gap by establishing rigorous baselines for more advanced methods, systematically evaluating how well models can generalize from few samples using five widely-adopted strategies: regularization, augmentation, prototypical networks, siamese networks, pre-training with fine-tuning. Our research investigates:

(1) Which strategies provide the strongest performance gains in few-shot graph learning scenarios
(2) How strategy effectiveness varies across different graph domains (citation networks, co-authorship graphs, molecular graphs)
(3) Whether performance improvement correlates more strongly with training sample size or learning strategy choice

Our comparative analysis across multiple graph domains provides crucial insights into which approaches offer the strongest inductive biases for different graph types and few-shot scenarios.

In recent years, numerous regularization techniques have been proposed for graph neural networks, most focusing on topological-level adjustment [Fang et al. 2023]. In the following we only summarize the implementations proposed in this article, pointing to the respective papers for further details.

Two common regularization techniques that help prevent overfitting and enhance generalization are Cosine Normalization (CSN) [Luo et al. 2018], also known as L2-constrained softmax [Ranjan et al. 2017], and entropy regularization (ER) [Grandvalet and Bengio 2004]. In CSN the input embedding is normalized to lie on a unit hypersphere along with the rows of the classification matrix. Hence, the computed logits can be seen as cosine similarities between the embedding vector and a representative vectors of the classes. This approach has proven to be effective in many scenarios [Luo et al. 2018], however some unexpected results are still under investigation [Steck et al. 2024]. On the other hand, ER is an explicit regularizer that adds a term to the standard cross-entropy loss to encourage higher-entropy predictions. Penalizing overconfidence can improve the final performance [Pereyra et al. 2017] and is suitable when limited data are available.

Augmentation refers to any technique that expands the dataset, transforming the input data without changing the labels. Inspired by [Fang et al. 2023], we implement several dropout strategies which modify the graph topology: DropNode randomly discards some node, DropEdge randomly discards some edge, DropAttributes partially masks the node embeddings, DropMessage partially masks the message passed to a neighbor node. In addition, two augmentation strategies have been tested, namely edge augmentation, which randomly adds some edge to the input graph, and rewiring, which modifies the topology of the input graph according to a heuristics.

Prototypical networks aim to compute a vector representation for each considered class, called prototype, and assign new samples to the class of the closest representative in the embedding space. Despite its simplicity, this method has proved to be particularly effective in few-shot classification [Snell et al. 2017].

Siamese networks aim to compute similarity functions and cope well with FSL [Koch et al. 2015]. To achieve this goal, multiple samples are passed through the same network followed by a contrastive loss function, which pushes embeddings belonging to the same class closer together and others further apart. In this work, we experiment with pairwise loss (PW) and triplet loss (TPL).

Pretraining and fine-tuning is inspired by [You et al. 2021], where a network is pre-trained in an unsupervised way to learn robust representations on graph-structured data. Specifically, the network is initialized to produce similar embedding representations to different augmented versions of the same graph and then trained for classification.

Our experiments use a consistent three-layer graph convolutional architecture across all strategies, with systematically increasing training samples (5-80 per class). Performance is evaluated using micro-accuracy and F1-scores averaged over multiple independent runs, with improvements measured relative to a cross-entropy baseline model.

Our systematic comparison across diverse graph datasets reveals how strategy effectiveness varies with graph structure and complexity. Main outcomes are:

(1) a comprehensive empirical benchmark of five strategies for few-shot learning on graphs across nine diverse datasets;
(2) identification of domain-specific patterns showing that prototypical networks excel on datasets with well-separated class structures (like in Mutag dataset), graph rewiring strategies better preserve essential topological information in sparse citation networks while injecting some noise, and Cosine Normalization provides stronger performance benefits for denser structures like coauthorship networks and the Amazon Computers dataset by effectively constraining the embedding space;
(3) demonstration that these straightforward strategies significantly improve performance without requiring complex architectural modifications or meta-learning frameworks.

By validating these targeted hypotheses across different graph domains, our work not only establishes performance baselines but also provides practical insights into which approaches are most suitable for specific graph types in few-shot learning scenarios.

Organization of This Article. The remainder of this article is organized as follows. Section 2 mentions other surveys about FSL on graphs, while Section 3 outlines required background on GNNs. Method and experiments are described in Section 4 and 5, respectively; while the results are analysed in Section 6. Section 7 summarizes this article.

GitHub repo. NicolaMaestri00/FSL-on-GNNs

2 Related works

While there exists extensive literature on each considered technique, systematic comparisons across all these strategies remain scarce. For a comprehensive review of few-shot learning on graphs, we refer to Zhang et al. [Zhang et al. 2022], which categorizes approaches into metric-based and optimization-based methods. Our methodological choice of using a simple graph convolutional network is supported by Luo et al. [Luo et al. 2024], who demonstrated that GCN-based models can outperform complex architectures. For graph regularization techniques, Fang et al. [Fang et al. 2023] provide an overview of topological-level approaches, while Wang et al. [Zhao et al. 2023] catalog graph augmentation methods across node, edge, and feature levels. In the metric learning domain, Koch et al. [Koch et al. 2015] and Snell et al. [Snell et al. 2017] established foundational work for Siamese and Prototypical networks respectively, though their systematic application to graph data remains underexplored. Recent works like Tan et al. [Tan et al. 2021] for Graph Few-shot Class-incremental Learning and Huang et al. [Huang et al. 2020] for local interpretable model explanation have made progress on new techniques, but typically focus on single approaches rather than comparative analyses. Our work bridges this gap by systematically evaluating multiple strategies across diverse graph datasets, establishing which methods are most effective for specific graph domains.

3 Background

3.1 Notation

The notation used throughout the article is listed in Table 1.

3.2 Graph Neural Networks

Graph representation learning is a broad research area which aims to automate the encoding of nodes, edges or the whole graph into meaningful vector representation. Recent advancements in the field mostly rely on graph neural networks (GNNs), concerning which a detailed introduction can be found here [Wu et al. 2020]. GNNs learn node embedding via message passing:

$$h_v^{(l+1)} = COM\left(h_v^{(l)}, \left[AGG\left(\left\{h_u^{(l)} | \forall u \in \mathcal{N}_v\right)\right\}\right]\right) \tag{1}$$

where AGG and COM are the neighbor aggregation and combination functions, respectively. A representation for the whole graph can be obtained as a function of the node embeddings

$$h_G^{(l)} = READOUT\left\{h_v^{(l)} | \forall v \in V\right\} \tag{2}$$

3.3 Few-Shot Learning on Graphs

Few-shot learning refers to the ability of machine learning models to generalize from few training examples [Parnami and Lee 2022]. A common FSL setting considers a set of q labeled data for each

Table 1: Notation

Notations	Descriptions
G	A graph.
V	The set of nodes in a graph.
v	A node $v \in V$.
E	The set of edges in a graph.
e_{ij}	An edge $e_{ij} \in E$.
$\mathcal{N}(v)$	The neighbors of a node v.
$h_v^{(l)}$	The embedding of node v at l-th GNN layer.
AGG	Neighbor aggregation function.
COM	Neighbor combination function.
$\|\cdot\|$	Number of elements of a set.
$\|\|\cdot\|\|$	Euclidean distance
A	adiacency matrix for graph $\mathcal{G}$
$\tilde{A} = A + I_N$	adiacency matrix with self-loops
$\tilde{D_{ii}} = \sum_j A_{ij}$	
$W^{(l)}$	trainable weight matrix at layer l
$H^{(l)} \in \mathbb{R}^{N \times D}$	matrix of activations at layer l
$H^0 = X$	node features

one of k classes, also known as q-shots k-ways learning.
FSL strategies can be groupped into two main subcategories:

(1) metric-based methods: learn similarity metric between support data and query data
(2) optimization-based methods: learn to quickly adapt a prior knowledge to new few-shot task with gradient computation

In recent years, considerable effort has been made to combine graph representation learning with prominent FSL strategies since in most real situations few labeled data are available for graph-structured data. In this framework, two areas that have naturally emerged are FSNC and FSGC.

3.3.1 Few-Shot Node Classification. FSNC aims to develop a machine learning model able to correctly classify the nodes of a graph despite only a limited number of labeled nodes are available for each class. A standard approach consists in pretraining the model on classes with a lot of samples and adapting this knowledge to new classes with few samples. In our work, the model has access to the entire graph at training stage, including the test nodes, but only few training labels can be seen. This scenario is known in literature as transductive learning.

3.3.2 Few-Shot Graph Classification. FSGC aims to develop a machine learning model able to correctly classify entire graphs despite only a limited number of samples are available for each class. As in FSNC, a typical approach consists in adapting to the new classes a model already trained for a related task. However, in this case, the model must generalize to new graphs not seen at training stage, hence it employs inductive learning.

4 Method

4.1 Model Architecture

For both FSNC and FSGC, we use a three-layered graph convolutional network customized for each specific task. We implement the graph convolutional layer as proposed by Kipf and Welling [Kipf and Welling 2017]:

$$H^{(l+1)} = ReLU\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{3}$$

which can be seen at the node level as:

$$h_v^{(l+1)} = ReLU\left(W^{(l)T}\sum_{u \in v \cup \mathcal{N}_v}\frac{1}{\sqrt{|\mathcal{N}_v|+1}\sqrt{|\mathcal{N}_u|+1}}h_u^{(l)}\right) \tag{4}$$

For FSGC, we aggregate node embeddings using the mean pooling function as our readout operation:

$$h_G = \frac{1}{|V|}\sum^{|V|} h_v \tag{5}$$

The FSNC-model consists of an encoder made of two convolutional layers and a classifier which yields the logits over the classes. Notably, we implement the classifier as a graph convolutional layer, allowing the classification of a node to depend on both its embedding and those of its neighbors. The FSGC-model consists of an encoder and a linear classifier. The encoder stacks three convolutional layers to extract contextual information and then uses the pooling function to retrieve an embedding for the whole graph.

4.2 Data Preparation

For both tasks, we create training sets with incrementally increasing sample sizes of [5, 10, 20, 40, 60, 80] samples per class and use these consistently across all experiments. Specifically, for FSNC we sample training masks with these specified numbers of nodes per class while using the default masks for validation and evaluation. For FSGC, we first define the validation and test datasets and then create collections of training graphs for each class to ensure fair comparisons across all strategies.

4.3 Implemented Strategies

The strategies applied can be adapted to both tasks without major changes. Therefore, we present them together, highlighting specific adaptations when needed.

4.3.1 Baseline and Regularization Approaches. We start by training the basic model with cross-entropy loss and AdamW optimizer as our baseline. We then experiment with CSN by normalizing the embedding vectors and the rows of the classification matrix:

$$cos_{sim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}||_2||\mathbf{b}||_2} \tag{6}$$

For entropy regularization, we modify the loss function by adding a regularization term:

$$\mathcal{L} = CE + \lambda_{ER}\, ER \tag{7}$$

with

$$ER = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} p_{i,c}\log p_{i,c} \tag{8}$$

where $p_{i,c}$ represents the predicted probability of sample i belonging to class c. By using a negative λ_{ER}, we penalize overconfident predictions, encouraging the model to maintain higher uncertainty.

4.3.2 Augmentation Strategies. We implement various dropout-based augmentation strategies by adding specific dropout blocks to the model architecture:

- DropNode: masks a subset of nodes
- DropEdge: masks a subset of edges
- DropAttributes: masks a portion of the node embeddings
- DropMessage: masks the message passed to the neighbors

Additionally, we experiment with two graph-specific augmentation techniques: random augmentation, which adds edges to the input graph, and rewiring. We implement domain-specific rewiring based on the type of graph being processed. For molecular datasets used in graph classification, rewiring preserves the degree of each node, ensuring that the augmented graphs maintain chemical plausibility by respecting constraints on atomic bonds. For node classification on citation and coauthorship graphs, our rewiring preserves the number of outgoing edges and the class of the destination node.

4.3.3 Contrastive Learning Approaches. We experiment with contrastive loss functions to learn more robust representations for nodes and graphs. Specifically, we consider:

- pairwise loss

$$\mathcal{L}_{PW}(x_1, x_2, y) = y||x_1 - x_2|| + (1-y)max(0, m - ||x_1 - x_2||) \quad (9)$$

with

$$\begin{cases} y = 0, & \text{if } x_1, x_2 \text{ belong to the same class} \\ y = 1, & \text{otherwise} \end{cases}$$

and m a fixed margin

- triplet loss

$$\mathcal{L}_{TPL}(a, p, n) = max\{||a - p|| + m - ||a - n||\} \quad (10)$$

with p a sample with the same label as a, n a sample with different label from a, m a fixed margin

We implement these contrastive approaches within a siamese network architecture, training with a combination of cross-entropy and contrastive losses. With PW loss, we pass two samples through the network and then compute the PW loss between their embeddings and the CE loss for the classification of the first sample. With TPL, we pass an anchor sample along with a positive and a negative sample, then compute the TPL loss between the embeddings and the CE loss on the classification of the anchor embedding.

4.3.4 Pretraining and Prototypical Networks. Pretraining and fine-tuning is implemented in a straightforward way for both tasks: we choose an augmentation technique and pass two augmented versions of the same graph through the network along with another sample. Then we use a contrastive loss to push the embeddings of the augmented pair closer together while pushing the other sample's embedding farther apart. After this initialization, the model is fine-tuned with cross-entropy loss. For Prototypical Networks, we employ a different training procedure focused on learning class prototypes. These networks are trained solely with a contrastive loss to cluster samples belonging to the same class. At each step, we sample candidates for each class, compute the contrastive loss, and update the weights. Then we compute a prototype for each class and classify elements in the validation set by assigning them to the closest prototype in the embedding space. This process continues until the performance on the validation set begins to decrease, at which point we stop training to prevent overfitting.

5 Experiments

5.1 Datasets

We briefly describe the datasets used in our experiments. The details of these datasets are summarized in Table 2.

5.1.1 Mutag, Proteins, Enzymes. are three molecular datasets for graph classification of increasing complexity. Mutag is a binary-classification dataset of 188 chemical compounds labeled by their mutagenic effect on bacteria. Each compound is represented as a graph, with atoms as nodes and bonds as edges. Proteins is a binary-classification dataset containing 1113 molecules labeled as enzymes or non-enzymes. Each molecule is represented as a graph, with aminoacids as nodes linked by an edge if they are less than 6 Angstroms apart. Enzymes is a dataset containing 600 protein structures equally subdivided into 6 classes. Each protein is represented as a graph, with atoms as nodes and bonds as edges.

5.1.2 Cora, CiteSeer, PubMed. are three citation graphs for node classification of increasing complexity. The Cora dataset is a graph of 2708 scientific publications linked by 5429 citation edges, where each node belongs to one out of seven classes and is described by 1-hot vector of legth 1433 indicating the presence of a word from the Cora dictionary. The CiteSeer dataset is a graph of 3312 scientific publications linked by 4732 citation edges, where each node belongs to one out of six classes and is described by 1-hot vector of legth 3703 indicating the presence of a word from the CiteSeer dictionary. The PubMed dataset is a graph of 19,717 scientific publications linked by 44,338 citation edges, where each node belongs to one out of three classes and is described by a TF–IDF–weighted feature vector over a 500-word vocabulary.

5.1.3 Coauthor CS, Physics. are co-authorship graphs for node classification, where nodes are authors and a edge link two authors if they co-authored a paper. Nodes are described by a vector counting paper keywords and are labeled according to the field of study. Coauthor CS has 18,333 nodes, 163,788 edges, 15 classes, 6,805 node features; while Coauthor Physics has 34,493 nodes, 495,924 edges, 5 classes, 8,415 node features.

5.1.4 Amazon Computers. is a co-purchase graph extracted from Amazon, where nodes represent products, edges represent the co-purchased relations of products, and features are bag-of-words vectors extracted from product reviews.

Graph Classification Dataset	Type	n° of graphs	Classes	Features	Avg. Nodes	Avg. Edges	Avg. Edge Density
MUTAG	Molecular graph	188	2	7	17.93	19.79	0.0616
PROTEINS	Protein graph	1113	2	3	39.06	72.82	0.0478
ENZYMES	Protein graph	600	6	3	32.63	62.14	0.0584
Node Classification Dataset	**Type**	**n° of graphs**	**Classes**	**Features**	**Nodes**	**Edges**	**Edge density**
Cora	Citation	1	7	1433	2708	5429	0.0004
CiteSeer	Citation	1	6	3703	3312	4732	0.0004
PubMed	Citation	1	3	500	19717	44338	0.0001
Coauthor CS	Coauthorship	1	15	6805	18333	163788	0.0001
Coauthor Physics	Coauthorship	1	5	8415	34493	495924	0.0001
Amazon Computers	Co-purchase	1	10	767	13381	245778	0.0007

Table 2: Dataset statistics

5.2 Hyperparameters

Table 3 lists the hyperparameters used for training.

Hyperparameter	GC	NC
Runs	10	10
Epochs	500	500
Patience	25	25
Hidden channels	64	128
Learning rate	0.001	0.01
Weight decay	0.00005	0.0005
λ_{ER}	- 0.5	- 0.5
DropMess. rate	0.8	0.8
DropNode rate	0.3	0.3
DropEdge rate	0.3	0.3
DropAttr. rate	0.3	0.3
Augment. rate	0.3	0.3
Rewiring rate	0.3	0.3

Table 3: Hyperparameters for Graph and Node Classification

5.3 Evaluation

This experimental design allows us to systematically address our three research questions by comparing strategy performance across diverse graph domains. The molecular, citation and coauthorship graphs represent different structural patterns and application domains, enabling us to identify which strategies offer the strongest inductive biases for specific graph types. We run each experiment for 500 epochs with early stopping, setting patience to 25 epochs. We measure performance with micro-accuracy and F1-score, reporting an average over 10 runs with relative standard deviation. To foster reproducibility, we run the experiments inside a container based on nvidia image PyG:24.07. The details on how to run the code are specified in the github repo. As our preliminary analysis indicates, the benefit of each tested strategy is strictly related to the dataset characteristics. Section 6 presents a detailed analysis of how each strategy performed across these diverse datasets, highlighting the relative performance gains and domain-specific patterns that emerged from our experiments.

6 Results

6.1 Mutag Analysis

Table 4 reports micro-accuracy and F1-scores across four few-shot settings, with Figure 1 visualizing performance trends. Prototypical Networks consistently outperform all methods in low-sample settings, achieving 83.7 ± 0.5% accuracy at FS5 compared to 80.5 ± 1.5% for the baseline. This superior performance is accompanied by significantly lower standard deviations, indicating greater robustness to data variability, a crucial advantage in few-shot scenarios. Conversely, dropout strategies underperform at lower sample sizes (78.4 ± 2.1% at FS5), suggesting standard dropout rates may be too aggressive for molecular graphs with limited training data.

As sample size increases, strategy effectiveness shifts notably. While ProtoNet dominates in extreme few-shot settings (FS5-FS20), regularization approaches (CSR, ER) become increasingly competitive with more training data, with ER achieving the highest accuracy at FS40 (87.7 ± 0.7%). This indicates that explicitly controlling prediction confidence through regularization becomes more effective than prototype-based representations as training data increases. Meanwhile, Siamese networks offer modest improvements over the baseline but fail to match ProtoNet's performance in most settings, likely due to differences in handling outlier samples.

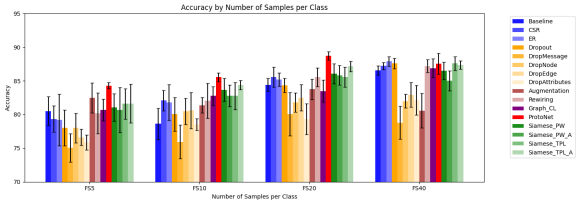

Figure 1: micro-accuracy ± std on the Mutag dataset

6.2 Proteins Analysis

Analysis of the Proteins results, with micro-accuracy and F1-scores detailed in Table 5 and micro-accuracy trends visualized in Figure 2, reveals distinct performance patterns for the evaluated strategies. Notably, methods that introduce structural noise, such as DropNode and DropEdge, generally demonstrate benefits over the Baseline. DropNode, for instance, achieves a significant micro-accuracy gain of approximately 1.5 % at FS10 (68.25 ± 0.36 compared to Baseline's 66.76 ± 0.80) and an even larger F1-score improvement of nearly 1.9% at the same sample size (67.86 ± 0.69 versus 65.96 ± 1.10). DropEdge tends to show increasing advantages at higher sample settings (FS20-FS80), delivering micro-accuracy gains of up to 1.06 % at FS20 and the highest F1-score gain of 1.56 % at FS80. These results suggest that injecting a degree of structural perturbation can aid model generalization on the more complex graph structures found in the Proteins dataset.

In contrast, several other strategies exhibited more varied performance on this dataset. Cosine Regularization (CSR) generally underperformed the Baseline across most few-shot settings in both micro-accuracy and F1-score. Prototypical Networks (ProtoNet) also struggled, particularly in the very low sample setting (e.g., FS5 micro-accuracy of 58.41 ± 1.86), indicating that its approach might be less effective for the complex class separations in Proteins compared to simpler datasets like Mutag. Entropy Regularization (ER) offered performance often comparable to or slightly below the Baseline. Standard data augmentation techniques and graph rewiring also largely underperformed in the lowest sample settings (FS5-FS20). However, graph rewiring notably excelled at the FS40 setting, achieving the top micro-accuracy (69.52 ± 0.57) and F1-score (68.52 ± 0.89) among all methods, suggesting its specific topological adjustments can be beneficial when a moderate amount of labeled data becomes available.

6.3 Enzymes Analysis

The Enzymes dataset presents a considerable challenge for few-shot graph classification, as evidenced by the micro-accuracy and F1-scores in Table 6 and the performance trends visualized in Figure 3.

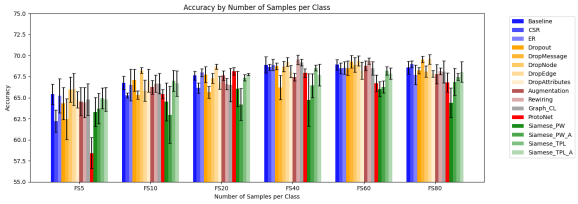

Figure 2: micro-accuracy ± std on the Proteins dataset

A striking overall observation is the limited success of all strategies in achieving high performance, with no method managing to elevate micro-accuracy or F1-scores above the 40%, even when using 80 training examples per class. Despite these general difficulties, Cosine Regularization (CSR) emerges in the lower to moderate sample settings. It consistently yields the best micro-accuracy and F1-scores at FS5 (28.67 ± 1.80 and 24.33 ± 1.63, respectively), FS10 (29.17 ± 0.75 and 26.77 ± 1.05, respectively) and again at FS40 (34.50 ± 2.15 and 32.24 ± 2.49, respectively). When more training data is available, specifically at the FS60 and FS80 levels, Graph Contrastive Learning (Graph_CL) emerges as the top performer, recording the highest micro-accuracy (e.g., 37.83 ± 3.71 at FS80) and F1-scores (e.g., 37.97 ± 4.08 at FS80).

The pervasive struggle to attain higher accuracy levels suggests that the base Graph Neural Network architecture faces significant hurdles in extracting sufficiently discriminative features from the complex graph structures inherent to the Enzymes dataset, especially under few-shot conditions. Many of the other evaluated approaches, including various dropout-based augmentations and Prototypical Networks, generally failed to provide consistent improvements over the baseline, often yielding comparable or lower results. This difficulty is further highlighted by the relatively high variance observed for many methods across different sample sizes, pointing to instability in the learning process. These findings collectively underscore that the Enzymes dataset is particularly arduous for the tested few-shot learning strategies, likely due to intrinsic limitation of the model and data structure complexity.

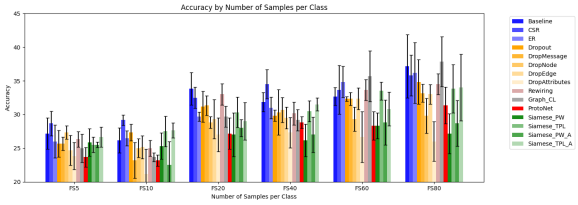

Figure 3: micro-accuracy ± std on the Enzymes dataset

6.4 Cora, CiteSeer, PubMed Analysis

The performance on citation networks shows distinct patterns across the three datasets as illustrated in Fig. 4. For Cora, topology modification strategies provide the most consistent benefits, with Rewiring achieving the highest micro-accuracy at FS5 (76.11%) and DropNode excelling with more training data (85.35% at FS80) as shown in Table 7. This suggests that augmentation in the sparse citation graph topology enhances learning from limited samples. Siamese networks with pairwise loss consistently outperform triplet loss variants, with Siamese_PW delivering the strongest performance at FS60 (85.03%), indicating effective similarity learning when moderate training data are available.

For CiteSeer, we observe more varied strategy effectiveness across sample sizes as reported in Table 8 and visualized in Fig. 4. Augmentation techniques provide remarkable gains in extremely low-data settings (58.47% at FS5 versus 53.26% baseline), while CSR excels at FS20 with a substantial improvement of 2.41 %. The inconsistent performance patterns across strategies suggest CiteSeer's structure requires different approaches at different sample sizes. Notably, ProtoNet significantly underperforms on both Cora and CiteSeer compared to molecular graphs, highlighting that on citation networks' structures prototype-based classification are less effective than in well-separated molecular datasets.

PubMed exhibits different behavior from the other citation networks, with the baseline unexpectedly outperforming all strategies in the lowest sample settings (FS5-FS10) as evident in Table 9 and Fig. 4. This suggests PubMed's larger graph size and more complex structure benefit from simpler models when very few samples are available. As sample size increases beyond 20 samples per class, dropout strategies, particularly DropMessage, become increasingly effective, achieving 82.02% F1 at FS60 compared to 81.07% of the baseline. This pattern indicates that injecting noise helps model generalization. Siamese_TPL_A demonstrates the strongest performance at FS80 (82.57% F1), showing that triplet-based contrastive learning becomes advantageous with larger training sets.

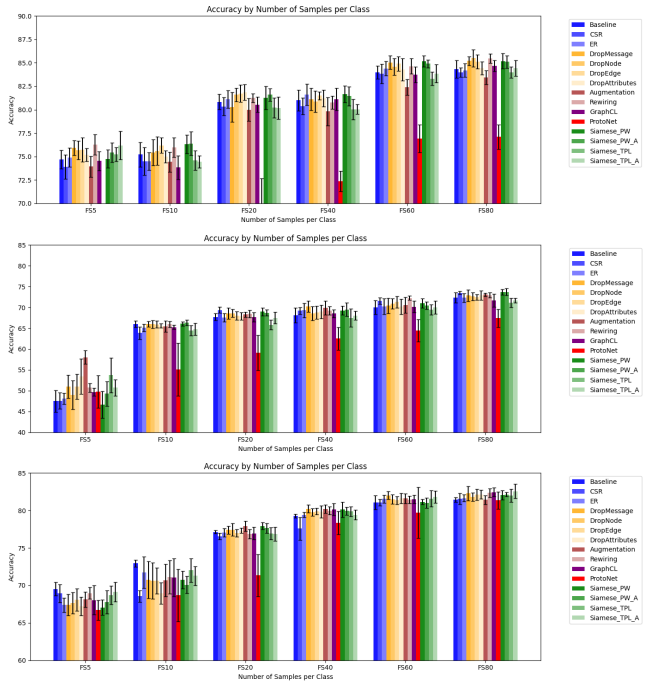

Figure 4: micro-accuracy ± std on Cora (top), CiteSeer (middle), PubMed (bottom)

6.5 Coauthors CS and Physics Analysis

Results listed in Table 10 and Table 11 reveal distinctive performance patterns across few-shot learning strategies on coauthorship networks, as also visible in Fig 5. Cosine Normalization (CSR) consistently outperforms other approaches, particularly in limited data scenarios with 1.53% accuracy improvement at FS5 on CS (86.85% vs baseline's 85.32%), and maintains its advantage even with increased training samples. Siamese networks with pairwise loss and augmentation excel at intermediate sample sizes (FS10 and FS40). Dropout-based strategies show varied effectiveness: DropAttributes achieves the highest accuracy at FS60 on CS (91.06%) and DropMessage delivers consistent gains in low-data scenarios on Physics (93.06% at FS5). Edge augmentation notably underperforms the baseline on CS, though rewiring shows moderate improvements. Prototypical Networks significantly underperform across both datasets, contrasting with their success on molecular graphs, achieving only 77.57% accuracy at FS5 on CS (nearly 8% below baseline) with persistently high variance across all sample sizes. This suggests that academic collaboration networks contain complex, overlapping class structures that resist prototype-based classification due to their high-dimensional, heterogeneous nature. The findings highlight an important domain-specific insight: strategies effective on molecular graphs with clear structural patterns may fail on citation and coauthorship networks where interdisciplinary boundaries create ambiguous class separations.

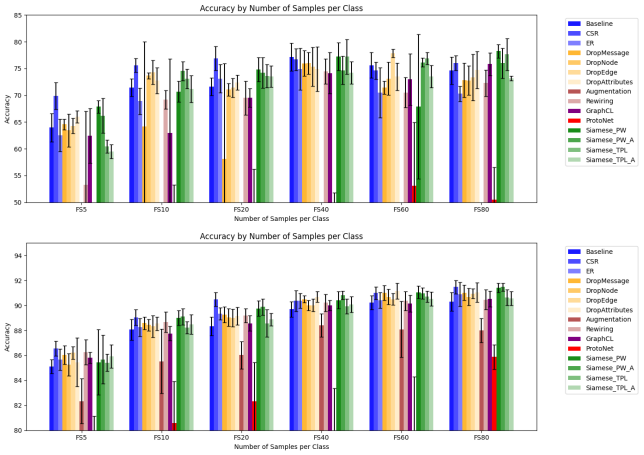

Figure 5: micro-accuracy ± std on CS (top) and Physics (bottom)

6.6 Amazon computers Analysis

Table 12 reports micro-accuracy and F1-scores on Amazon Computers dataset, showing distinct performance patterns across different few-shot learning strategies. As visible in Fig 6 Cosine normalization (CSR) consistently demonstrates superior performance in the most challenging few-shot scenarios (FS5, FS10, FS20), achieving gains of more than 4 % in both accuracy and F1-score compared to the baseline. This suggests that constraining the embedding space through normalization provides particularly strong benefits for this co-purchase network. Following CSR, Siamese networks with pairwise loss also show promising results, particularly as sample sizes increase. Notably, beyond FS40, performance gains appear to plateau across most strategies, with DropEdge achieving the highest accuracy (77.84 ± 0.80%) at FS60, while Siamese PW models at FS80 (78.24 ± 1.28%). This performance ceiling suggests the base GCN encoder may have reached its representational capacity under the current few-shot settings, and further improvements would likely require more expressive architectures or advanced meta-learning strategies. Additionally, some strategies show inconsistent performance: Augmentation consistently underperforms across all sample sizes, while ProtoNet struggles particularly with the Amazon dataset structure, achieving only 25.34 ± 3.68% accuracy at FS5, indicating this approach may not effectively capture the graph topology.

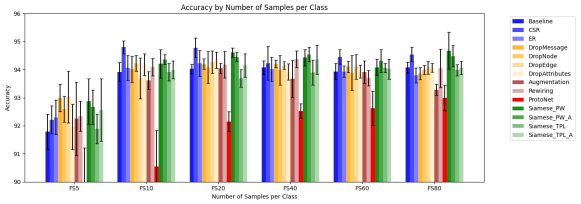

Figure 6: micro-accuracy ± std on Amazon Computers

7 Conclusion

Our systematic evaluation of regularization and few-shot learning strategies across diverse graph domains reveals that even simple three-layer GCN architectures can generalize effectively with limited labeled data, though strategy effectiveness is highly domain-dependent. For molecular graphs with well-separated class structures like Mutag, Prototypical Networks excel in extreme few-shot settings (5-20 samples per class), while topology-perturbation strategies such as graph rewiring and DropNode provide consistent benefits for sparse citation networks (Cora, CiteSeer, PubMed) by making the node representations more robust. In contrast, Cosine Normalization delivers the strongest performance improvements for coauthorship networks and Amazon Computers dataset by effectively constraining the embedding space. Despite these encouraging results, more sophisticated meta-learning frameworks, adaptive regularization techniques and cross-domain transfer methods remain necessary to achieve higher performance in extremely low-data settings, particularly for complex datasets like Enzymes where all tested approaches struggled. Our findings establish solid baselines for future research while demonstrating that even straightforward strategies can significantly improve performance without requiring complex architectural modifications or meta-learning frameworks, provided they are appropriately matched to the specific graph domain.

References

Taoran Fang, Zhiqing Xiao, Chunping Wang, Jiarong Xu, Xuan Yang, and Yang Yang. 2023. DropMessage: Unifying Random Dropping for Graph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 4 (June 2023), 4267–4275. https://doi.org/10.1609/aaai.v37i4.25545

Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised Learning by Entropy Minimization. In *Advances in Neural Information Processing Systems*, L. Saul, Y. Weiss, and L. Bottou (Eds.), Vol. 17. MIT Press, . https://proceedings.neurips.cc/paper_files/paper/2004file/96f2b50b5d3613adf9c27049b2a888c7Paper.pdf

Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, Dawei Yin, and Yi Chang. 2020. GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks. arXiv:2001.06216 [cs.LG] https://arxiv.org/abs/2001.06216

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs.LG] https://arxiv.org/abs/1609.02907

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, Vol. 2. Lille, ., ., 1–30.

Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. 2018. Cosine Normalization: Using Cosine Similarity Instead of Dot Product in Neural Networks. In *Artificial Neural Networks and Machine Learning – ICANN 2018*, Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis (Eds.). Springer International Publishing, Cham, 382–391.

Yuankai Luo, Lei Shi, and Xiao-Ming Wu. 2024. Classic GNNs are Strong Baselines: Reassessing GNNs for Node Classification. arXiv:2406.08993 [cs.LG] https://arxiv.org/abs/2406.08993

Archit Parnami and Minwoo Lee. 2022. Learning from Few Examples: A Summary of Approaches to Few-Shot Learning. arXiv:2203.04291 [cs.LG] https://arxiv.org/abs/2203.04291

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing Neural Networks by Penalizing Confident Output Distributions. https://openreview.net/forum?id=HkCjNI5ex

Rajeev Ranjan, Carlos D. Castillo, and Rama Chellappa. 2017. L2-constrained Softmax Loss for Discriminative Face Verification. arXiv:1703.09507 [cs.CV] https://arxiv.org/abs/1703.09507

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical Networks for Few-shot Learning. arXiv:1703.05175 [cs.LG] https://arxiv.org/abs/1703.05175

Harald Steck, Chaitanya Ekanadham, and Nathan Kallus. 2024. Is Cosine-Similarity of Embeddings Really About Similarity?. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24)*. ACM, ., 887–890. https://doi.org/10.1145/3589335.3651526

Zhen Tan, Kaize Ding, Ruocheng Guo, and Huan Liu. 2021. Graph Few-shot Class-incremental Learning. arXiv:2112.12819 [cs.LG] https://arxiv.org/abs/2112.12819

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2021. Graph Contrastive Learning with Augmentations. arXiv:2010.13902 [cs.LG] https://arxiv.org/abs/2010.13902

Chuxu Zhang, Kaize Ding, Jundong Li, Xiangliang Zhang, Yanfang Ye, Nitesh V. Chawla, and Huan Liu. 2022. Few-Shot Learning on Graphs. arXiv:2203.09308 [cs.LG] https://arxiv.org/abs/2203.09308

Tong Zhao, Wei Jin, Yozen Liu, Yingheng Wang, Gang Liu, Stephan Günnemann, Neil Shah, and Meng Jiang. 2023. Graph Data Augmentation for Graph Machine Learning: A Survey. arXiv:2202.08871 [cs.LG] https://arxiv.org/abs/2202.08871

Appendix

Method	FS5	FS10	FS20	FS40
Baseline	80.46 ± 1.46	79.44 ± 1.54	84.44 ± 0.81	86.20 ± 0.77
CSR	79.81 ± 2.51	82.13 ± 1.85	85.28 ± 1.40	87.04 ± 0.59
ER	80.19 ± 2.35	82.59 ± 1.23	85.00 ± 0.91	**87.69 ± 0.72**
Dropout	78.43 ± 2.11	79.72 ± 2.32	84.35 ± 1.05	87.31 ± 0.83
DropMessage	78.89 ± 1.93	78.89 ± 1.59	81.48 ± 2.19	80.19 ± 1.45
DropNode	79.07 ± 1.61	81.76 ± 0.72	82.31 ± 1.27	81.76 ± 0.59
DropEdge	79.72 ± 2.09	82.04 ± 1.86	83.43 ± 1.52	83.98 ± 1.25
DropAttributes	79.35 ± 1.38	78.52 ± 1.54	80.46 ± 2.57	82.22 ± 1.48
Augmentation	82.31 ± 1.68	80.74 ± 1.54	83.70 ± 1.45	80.56 ± 2.03
Rewiring	81.57 ± 0.97	82.87 ± 1.56	85.37 ± 1.42	86.94 ± 0.97
Graph_CL	80.37 ± 2.14	82.59 ± 1.48	83.24 ± 1.40	86.57 ± 1.45
ProtoNet	**83.70 ± 0.61**	**84.63 ± 0.74**	**88.52 ± 0.74**	87.04 ± 1.76
Siamese_PW	82.13 ± 0.93	83.61 ± 2.19	85.83 ± 1.44	86.11 ± 1.60
Siamese_PW_A	82.04 ± 1.99	82.59 ± 1.23	85.56 ± 1.45	84.63 ± 1.45
Siamese_TPL	81.85 ± 1.26	82.59 ± 1.70	85.19 ± 1.43	87.41 ± 1.03
Siamese_TPL_A	82.22 ± 1.08	83.80 ± 0.75	86.85 ± 0.81	87.13 ± 0.65

(a) Micro Accuracy

Method	FS5	FS10	FS20	FS40
Baseline	80.52 ± 2.15	78.61 ± 2.32	84.40 ± 0.95	86.57 ± 0.67
CSR	79.33 ± 1.95	82.08 ± 1.51	85.54 ± 1.48	87.21 ± 0.51
ER	79.19 ± 3.83	81.77 ± 2.66	85.17 ± 0.98	**87.89 ± 0.69**
Dropout	78.02 ± 2.66	80.04 ± 2.50	84.35 ± 1.05	87.58 ± 0.77
DropMessage	75.05 ± 2.09	75.93 ± 2.51	80.08 ± 3.21	78.79 ± 2.43
DropNode	77.99 ± 2.17	80.48 ± 1.94	81.79 ± 1.42	82.00 ± 1.03
DropEdge	76.62 ± 1.19	80.60 ± 2.70	82.50 ± 1.92	82.92 ± 1.84
DropAttributes	75.88 ± 1.11	78.50 ± 0.88	79.31 ± 2.27	82.14 ± 2.20
Augmentation	82.46 ± 2.22	81.38 ± 1.14	83.74 ± 1.54	80.58 ± 2.56
Rewiring	80.19 ± 3.05	82.05 ± 2.59	85.55 ± 1.39	87.20 ± 0.94
Graph_CL	80.66 ± 1.62	82.77 ± 1.40	83.49 ± 1.64	86.89 ± 1.42
ProtoNet	**84.28 ± 0.48**	**85.56 ± 0.65**	**88.71 ± 0.62**	87.54 ± 1.56
Siamese_PW	81.05 ± 2.02	83.66 ± 1.69	86.04 ± 1.48	86.49 ± 1.31
Siamese_PW_A	80.67 ± 3.32	82.76 ± 1.60	85.83 ± 1.47	85.00 ± 1.48
Siamese_TPL	81.58 ± 2.24	82.75 ± 1.98	85.54 ± 1.48	87.59 ± 1.01
Siamese_TPL_A	81.62 ± 2.87	84.38 ± 0.67	87.15 ± 0.79	87.36 ± 0.62

(b) Micro F1 Score

Table 4: Mutag dataset results: (a) Micro Accuracy and (b) Micro F1 across few-shot splits

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	65.40 ± 1.23	66.76 ± 0.80	67.63 ± 0.53	68.90 ± 0.95	68.92 ± 0.61	68.57 ± 0.77
CSR	62.23 ± 1.30	65.26 ± 0.29	66.14 ± 0.62	68.62 ± 0.36	68.48 ± 0.63	68.97 ± 0.40
ER	65.22 ± 2.03	66.53 ± 1.88	67.98 ± 0.46	68.90 ± 0.68	68.53 ± 0.74	67.72 ± 1.18
Dropout	64.32 ± 1.91	67.08 ± 1.30	67.75 ± 0.93	68.74 ± 0.40	68.51 ± 0.86	68.23 ± 0.37
DropMessage	62.46 ± 2.44	65.33 ± 0.49	65.66 ± 0.72	66.21 ± 1.42	69.26 ± 0.74	69.54 ± 0.38
DropNode	65.95 ± 1.48	**68.25 ± 0.36**	67.31 ± 0.59	68.71 ± 0.64	68.90 ± 0.86	68.11 ± 0.66
DropEdge	**66.00 ± 1.89**	65.84 ± 1.28	**68.69 ± 0.30**	69.24 ± 0.51	**69.36 ± 0.59**	**69.56 ± 0.58**
DropAttributes	64.76 ± 0.98	67.01 ± 1.36	66.78 ± 0.75	68.11 ± 0.71	68.21 ± 1.03	67.84 ± 0.39
Augmentation	64.55 ± 1.64	66.25 ± 0.81	67.63 ± 0.58	67.45 ± 0.46	68.80 ± 0.57	67.79 ± 1.14
Rewiring	64.44 ± 1.79	66.67 ± 1.02	66.62 ± 0.66	**69.52 ± 0.57**	69.36 ± 0.36	68.16 ± 0.39
Graph_CL	64.78 ± 1.87	66.67 ± 1.21	66.53 ± 1.98	69.20 ± 0.40	68.48 ± 0.78	68.07 ± 1.30
ProtoNet	58.41 ± 1.86	65.43 ± 0.56	68.14 ± 0.46	67.93 ± 0.53	66.71 ± 0.99	66.80 ± 1.13
Siamese_PW	63.29 ± 1.75	64.51 ± 2.25	66.05 ± 2.11	64.74 ± 3.12	66.02 ± 0.89	64.39 ± 1.75
Siamese_PW_A	63.70 ± 1.81	62.97 ± 3.38	64.21 ± 1.89	66.44 ± 1.39	66.28 ± 0.71	66.87 ± 1.60
Siamese_TPL	64.94 ± 1.24	66.99 ± 1.23	67.36 ± 0.38	68.53 ± 0.33	68.18 ± 0.49	67.45 ± 0.42
Siamese_TPL_A	64.78 ± 1.41	66.69 ± 1.51	67.77 ± 0.12	67.68 ± 1.29	67.86 ± 0.67	68.05 ± 1.21

(a) Micro Accuracy

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	64.01 ± 0.84	65.96 ± 1.10	66.61 ± 0.41	67.60 ± 1.30	67.18 ± 0.53	67.16 ± 1.16
CSR	57.01 ± 0.74	63.98 ± 0.39	64.47 ± 0.85	67.33 ± 0.41	67.20 ± 0.79	67.50 ± 0.39
ER	63.65 ± 1.43	65.79 ± 1.86	66.82 ± 0.96	67.75 ± 1.61	67.24 ± 0.59	66.91 ± 1.14
Dropout	62.75 ± 1.96	66.80 ± 1.24	66.73 ± 0.61	67.63 ± 1.03	67.30 ± 1.11	67.13 ± 0.39
DropMessage	55.24 ± 5.37	63.94 ± 0.42	64.26 ± 0.71	64.36 ± 1.45	67.77 ± 0.38	68.18 ± 0.26
DropNode	**64.36 ± 2.76**	**67.86 ± 0.69**	66.53 ± 0.81	67.35 ± 0.66	68.43 ± 0.91	67.01 ± 0.53
DropEdge	63.78 ± 3.63	64.55 ± 1.66	**67.68 ± 0.46**	67.61 ± 0.55	**68.66 ± 0.64**	**68.72 ± 0.83**
DropAttributes	61.07 ± 3.35	66.54 ± 1.61	65.97 ± 1.10	67.25 ± 0.96	67.06 ± 1.26	66.56 ± 0.58
Augmentation	62.93 ± 2.03	65.75 ± 0.83	66.10 ± 1.36	65.80 ± 1.56	67.26 ± 1.02	65.76 ± 1.28
Rewiring	63.25 ± 1.21	65.83 ± 1.23	66.05 ± 0.63	**68.52 ± 0.89**	68.04 ± 0.63	66.89 ± 0.65
Graph_CL	62.56 ± 1.21	66.46 ± 1.01	65.67 ± 1.70	68.32 ± 0.35	67.38 ± 1.39	66.91 ± 1.65
ProtoNet	58.57 ± 1.76	65.21 ± 0.56	67.63 ± 0.72	67.78 ± 0.43	66.68 ± 0.93	66.75 ± 1.20
Siamese_PW	62.10 ± 1.06	63.65 ± 2.74	65.42 ± 1.92	63.78 ± 2.76	64.58 ± 0.78	63.87 ± 1.74
Siamese_PW_A	62.68 ± 2.76	61.62 ± 2.95	63.28 ± 2.21	65.18 ± 1.94	65.53 ± 0.58	65.78 ± 1.70
Siamese_TPL	62.68 ± 2.10	66.56 ± 1.14	66.58 ± 0.59	67.67 ± 0.36	67.12 ± 0.36	66.39 ± 0.45
Siamese_TPL_A	63.03 ± 1.10	65.78 ± 1.74	66.32 ± 0.88	66.49 ± 1.89	66.40 ± 0.92	66.89 ± 1.48

(b) Micro F1 Score

Table 5: Proteins dataset results: (a) Micro Accuracy and (b) Micro F1 across few-shot splits

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	27.17 ± 2.33	26.17 ± 1.87	**33.83 ± 2.39**	31.83 ± 1.43	32.67 ± 1.33	37.17 ± 4.67
CSR	**28.67 ± 1.80**	**29.17 ± 0.75**	32.50 ± 1.58	**34.50 ± 2.15**	33.67 ± 3.64	35.83 ± 3.03
ER	26.00 ± 2.44	26.50 ± 1.11	29.67 ± 0.67	31.00 ± 1.62	34.83 ± 2.32	36.17 ± 4.52
Dropout	25.67 ± 2.00	27.33 ± 1.22	31.17 ± 2.27	29.83 ± 0.82	32.33 ± 0.33	34.83 ± 3.31
DropMessage	25.67 ± 0.97	23.17 ± 2.66	31.33 ± 1.45	30.33 ± 3.36	32.33 ± 0.97	33.17 ± 1.33
DropNode	27.33 ± 0.97	25.00 ± 1.39	28.83 ± 0.85	30.67 ± 1.86	29.33 ± 1.78	29.83 ± 2.60
DropEdge	24.67 ± 2.21	25.17 ± 1.70	29.33 ± 2.91	29.50 ± 1.63	32.33 ± 1.62	33.00 ± 1.45
DropAttributes	23.83 ± 2.01	21.17 ± 3.71	27.00 ± 2.51	27.33 ± 2.26	26.67 ± 3.76	26.00 ± 2.95
Rewiring	26.33 ± 1.13	25.00 ± 1.18	33.00 ± 1.55	30.17 ± 1.70	33.67 ± 1.55	34.50 ± 1.55
Graph_CL	25.00 ± 2.11	23.67 ± 0.67	29.67 ± 1.55	29.17 ± 1.58	**35.67 ± 3.78**	**37.83 ± 3.71**
ProtoNet	23.67 ± 1.45	23.17 ± 0.82	27.17 ± 2.33	28.83 ± 0.85	28.33 ± 2.11	31.33 ± 2.72
Siamese_PW	25.83 ± 2.04	25.33 ± 1.94	27.00 ± 3.23	26.17 ± 2.39	28.33 ± 1.90	27.17 ± 2.96
Siamese_TPL	25.50 ± 1.13	27.50 ± 2.24	30.33 ± 2.27	30.50 ± 1.55	33.50 ± 1.33	33.83 ± 3.60
Siamese_PW_A	25.50 ± 0.41	22.50 ± 3.46	28.00 ± 1.25	27.00 ± 2.61	28.83 ± 3.32	28.67 ± 3.40
Siamese_TPL_A	26.67 ± 1.49	27.67 ± 1.11	29.00 ± 2.76	31.50 ± 0.97	30.83 ± 2.53	34.00 ± 4.96

(a) Micro Accuracy

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	23.66 ± 2.34	23.44 ± 2.25	**32.28 ± 2.81**	28.83 ± 1.79	29.51 ± 2.64	36.53 ± 5.04
CSR	**24.33 ± 1.63**	**26.77 ± 1.05**	30.86 ± 1.88	**32.24 ± 2.49**	31.00 ± 3.94	33.67 ± 3.29
ER	22.94 ± 1.95	23.58 ± 2.74	27.52 ± 0.91	29.51 ± 0.97	32.68 ± 2.08	34.56 ± 5.49
Dropout	24.04 ± 2.57	25.04 ± 1.58	29.76 ± 2.23	28.10 ± 1.28	30.66 ± 1.41	32.63 ± 4.40
DropMessage	20.34 ± 1.21	19.34 ± 3.96	28.37 ± 3.57	26.81 ± 3.48	29.16 ± 0.91	30.26 ± 1.19
DropNode	22.41 ± 2.98	20.97 ± 1.79	25.69 ± 1.70	29.30 ± 2.19	27.10 ± 2.11	25.69 ± 2.46
DropEdge	18.22 ± 3.28	21.14 ± 1.82	27.98 ± 3.13	25.01 ± 1.68	28.25 ± 1.25	29.57 ± 2.62
DropAttributes	20.95 ± 1.70	17.23 ± 4.01	24.51 ± 1.95	23.22 ± 2.14	22.45 ± 4.66	20.13 ± 2.74
Rewiring	23.97 ± 1.33	21.49 ± 1.89	31.52 ± 2.29	27.67 ± 2.14	31.87 ± 1.85	33.12 ± 1.78
Graph_CL	22.58 ± 2.50	22.17 ± 0.77	29.23 ± 1.72	27.82 ± 1.03	**35.05 ± 3.75**	**37.97 ± 4.08**
ProtoNet	21.69 ± 1.74	22.41 ± 0.90	26.38 ± 2.34	26.99 ± 0.77	27.29 ± 1.97	29.38 ± 2.71
Siamese_PW	22.41 ± 2.57	22.37 ± 2.61	24.73 ± 2.74	22.92 ± 4.06	26.75 ± 2.43	23.70 ± 2.90
Siamese_TPL	22.66 ± 1.16	25.55 ± 2.81	28.17 ± 2.96	29.12 ± 2.01	31.23 ± 1.57	32.21 ± 3.36
Siamese_PW_A	21.25 ± 2.03	19.50 ± 4.18	25.21 ± 1.23	23.00 ± 3.19	25.03 ± 3.77	25.32 ± 3.23
Siamese_TPL_A	23.97 ± 1.26	26.52 ± 1.20	27.51 ± 2.34	28.68 ± 1.82	29.31 ± 2.38	31.35 ± 5.77

(b) Micro F1 Score

Table 6: Enzymes dataset results: (a) Micro Accuracy and (b) Micro F1 across few-shot splits

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	74.50 ± 0.99	74.97 ± 1.34	80.63 ± 0.81	80.87 ± 1.11	83.73 ± 0.68	84.14 ± 1.03
CSR	73.62 ± 1.29	74.24 ± 1.44	80.18 ± 0.95	80.30 ± 0.84	83.73 ± 1.02	83.88 ± 0.51
ER	74.67 ± 1.02	74.21 ± 1.05	80.91 ± 0.95	81.51 ± 1.17	84.21 ± 0.71	83.99 ± 0.69
DropMessage	75.75 ± 0.83	75.18 ± 1.39	80.13 ± 1.60	80.99 ± 1.20	84.84 ± 0.73	85.06 ± 0.47
DropNode	75.52 ± 0.99	75.47 ± 1.55	81.45 ± 0.72	80.74 ± 1.13	84.40 ± 0.92	85.35 ± 0.99
DropEdge	75.54 ± 1.32	75.92 ± 0.83	81.60 ± 0.98	81.33 ± 0.41	84.66 ± 0.83	84.95 ± 0.77
DropAttributes	75.01 ± 0.70	74.70 ± 0.68	**81.68 ± 0.98**	81.08 ± 0.92	84.08 ± 1.17	84.24 ± 0.70
Augmentation	73.67 ± 1.10	74.27 ± 1.04	79.88 ± 1.22	79.72 ± 1.47	82.28 ± 0.88	83.27 ± 0.76
Rewiring	**76.11 ± 1.11**	75.78 ± 1.04	81.02 ± 0.49	80.62 ± 0.71	84.46 ± 0.87	**85.27 ± 0.61**
GraphCL	74.27 ± 0.97	73.71 ± 1.27	80.34 ± 0.77	80.94 ± 1.22	83.53 ± 0.85	84.47 ± 0.58
ProtoNet	58.83 ± 3.94	59.80 ± 2.44	69.74 ± 2.83	72.14 ± 1.14	76.72 ± 1.45	76.88 ± 1.31
Siamese_PW	74.41 ± 0.98	76.11 ± 1.07	81.16 ± 1.24	**81.52 ± 0.92**	**85.03 ± 0.61**	85.05 ± 0.87
Siamese_PW_A	75.16 ± 1.18	**76.12 ± 1.25**	81.41 ± 0.65	81.31 ± 1.05	84.75 ± 0.43	84.96 ± 0.68
Siamese_TPL	75.08 ± 0.75	74.19 ± 1.22	80.04 ± 1.04	79.89 ± 1.14	83.05 ± 0.72	83.81 ± 0.61
Siamese_TPL_A	76.00 ± 1.54	74.10 ± 0.76	79.96 ± 1.24	79.90 ± 0.51	83.60 ± 1.07	84.23 ± 0.86

(a) Micro Accuracy

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	74.69 ± 0.97	75.21 ± 1.30	80.84 ± 0.80	81.00 ± 1.12	83.98 ± 0.68	84.34 ± 0.94
CSR	73.90 ± 1.27	74.48 ± 1.48	80.34 ± 0.97	80.37 ± 0.89	83.82 ± 1.04	83.98 ± 0.51
ER	74.89 ± 1.01	74.49 ± 0.92	81.12 ± 0.93	81.59 ± 1.14	84.40 ± 0.75	84.19 ± 0.71
DropMessage	75.93 ± 0.79	75.43 ± 1.39	80.28 ± 1.58	81.11 ± 1.19	85.03 ± 0.71	85.20 ± 0.46
DropNode	75.65 ± 0.99	75.59 ± 1.51	81.60 ± 0.72	80.86 ± 1.13	84.58 ± 0.90	**85.50 ± 0.93**
DropEdge	75.72 ± 1.29	76.18 ± 0.82	**81.73 ± 0.93**	81.49 ± 0.44	84.90 ± 0.75	85.12 ± 0.74
DropAttributes	75.16 ± 0.69	74.97 ± 0.63	81.86 ± 0.85	81.20 ± 0.90	84.27 ± 1.18	84.43 ± 0.68
Augmentation	73.93 ± 1.11	74.42 ± 1.07	80.00 ± 1.20	79.82 ± 1.51	82.40 ± 0.85	83.45 ± 0.73
Rewiring	**76.27 ± 1.10**	75.96 ± 1.04	81.24 ± 0.47	80.75 ± 0.69	84.64 ± 0.81	85.46 ± 0.51
GraphCL	74.53 ± 1.02	73.83 ± 1.27	80.54 ± 0.81	81.12 ± 1.17	83.73 ± 0.85	84.67 ± 0.58
ProtoNet	59.39 ± 4.13	60.31 ± 2.57	69.89 ± 2.75	72.35 ± 1.09	76.92 ± 1.49	77.10 ± 1.31
Siamese_PW	74.75 ± 0.96	76.34 ± 0.99	81.28 ± 1.24	**81.66 ± 0.90**	**85.16 ± 0.58**	85.18 ± 0.81
Siamese_PW_A	75.42 ± 1.05	**76.39 ± 1.26**	81.59 ± 0.65	81.44 ± 1.03	84.92 ± 0.42	85.11 ± 0.67
Siamese_TPL	75.22 ± 0.74	74.57 ± 1.03	80.22 ± 1.06	80.02 ± 1.10	83.31 ± 0.71	83.99 ± 0.60
Siamese_TPL_A	76.19 ± 1.53	74.43 ± 0.64	80.18 ± 1.18	80.03 ± 0.52	83.85 ± 0.95	84.44 ± 0.83

(b) Micro F1 Score

Table 7: Cora dataset results: (a) Micro Accuracy and (b) Micro F1 across few-shot splits

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	53.26 ± 2.42	65.41 ± 0.91	67.09 ± 0.94	68.37 ± 2.05	69.74 ± 1.80	72.55 ± 1.12
CSR	53.00 ± 0.98	63.52 ± 1.46	**69.50 ± 0.90**	69.35 ± 1.04	71.90 ± 0.91	73.75 ± 0.44
ER	54.06 ± 1.10	64.72 ± 0.98	67.15 ± 1.22	69.17 ± 2.07	70.44 ± 2.10	73.08 ± 1.10
DropMessage	55.05 ± 1.43	65.77 ± 0.70	68.23 ± 1.45	**70.63 ± 1.31**	70.63 ± 2.09	73.33 ± 1.57
DropNode	53.76 ± 1.36	65.82 ± 1.31	67.92 ± 1.24	68.50 ± 1.92	71.27 ± 1.45	73.03 ± 0.97
DropEdge	54.94 ± 1.57	65.74 ± 1.10	67.70 ± 0.91	68.95 ± 1.77	71.85 ± 1.78	72.82 ± 1.11
DropAttributes	56.16 ± 2.65	65.28 ± 0.44	67.59 ± 0.76	68.91 ± 1.96	70.33 ± 1.99	73.17 ± 1.31
Augmentation	**58.47 ± 1.72**	65.55 ± 1.33	68.09 ± 0.83	70.31 ± 1.91	71.00 ± 1.87	73.22 ± 0.41
Rewiring	54.95 ± 1.08	65.75 ± 1.23	68.29 ± 0.67	69.39 ± 1.09	**72.49 ± 0.62**	72.96 ± 0.76
GraphCL	54.55 ± 0.67	64.77 ± 0.80	67.10 ± 1.54	68.59 ± 1.12	70.24 ± 1.54	72.08 ± 1.44
ProtoNet	49.67 ± 3.78	54.80 ± 5.73	58.58 ± 4.02	62.34 ± 2.53	64.47 ± 2.63	67.15 ± 2.13
Siamese_PW	52.90 ± 1.48	65.82 ± 0.53	68.50 ± 1.12	69.56 ± 1.30	71.47 ± 1.32	**73.88 ± 0.79**
Siamese_PW_A	53.89 ± 1.40	**66.44 ± 0.72**	68.45 ± 0.80	69.81 ± 1.90	70.76 ± 0.98	73.74 ± 1.11
Siamese_TPL	57.36 ± 2.51	63.95 ± 1.22	65.43 ± 1.03	67.30 ± 2.65	69.56 ± 1.34	71.23 ± 1.63
Siamese_TPL_A	55.06 ± 0.79	64.49 ± 1.79	67.20 ± 1.53	68.32 ± 1.48	69.98 ± 1.66	71.94 ± 1.11

(a) Micro Accuracy

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	47.49 ± 2.66	66.01 ± 0.78	67.73 ± 0.84	68.17 ± 1.78	70.01 ± 1.65	72.32 ± 1.21
CSR	47.55 ± 1.96	63.85 ± 1.47	**69.39 ± 0.73**	69.20 ± 0.86	71.56 ± 0.77	73.49 ± 0.37
ER	48.08 ± 1.36	65.17 ± 0.89	67.57 ± 1.05	69.34 ± 1.72	70.24 ± 1.87	72.30 ± 1.09
DropMessage	50.95 ± 2.79	66.01 ± 0.71	68.61 ± 1.33	**70.20 ± 1.33**	70.43 ± 1.82	72.87 ± 1.42
DropNode	48.98 ± 3.44	65.96 ± 0.91	68.59 ± 0.96	68.57 ± 1.70	70.94 ± 1.24	72.64 ± 0.95
DropEdge	50.99 ± 3.02	65.93 ± 0.81	68.05 ± 1.01	68.77 ± 1.58	71.34 ± 1.40	72.46 ± 0.69
DropAttributes	53.45 ± 4.24	65.61 ± 0.50	67.92 ± 0.88	68.98 ± 1.61	70.21 ± 1.86	72.92 ± 1.07
Augmentation	**58.04 ± 1.65**	65.49 ± 1.32	68.31 ± 0.62	69.88 ± 1.68	70.59 ± 1.86	73.09 ± 0.35
Rewiring	50.71 ± 1.09	65.96 ± 0.73	68.49 ± 0.87	69.20 ± 1.00	**72.28 ± 0.50**	72.96 ± 0.64
GraphCL	49.68 ± 0.92	65.27 ± 0.52	67.69 ± 1.13	68.54 ± 0.95	70.15 ± 1.39	71.70 ± 1.59
ProtoNet	49.78 ± 3.92	55.08 ± 6.33	59.07 ± 4.23	62.51 ± 2.74	64.45 ± 2.63	67.40 ± 2.19
Siamese_PW	46.64 ± 3.18	66.07 ± 0.48	68.96 ± 0.90	69.27 ± 1.08	71.00 ± 1.10	73.65 ± 0.65
Siamese_PW_A	49.27 ± 2.96	**66.36 ± 0.66**	68.72 ± 0.69	69.51 ± 1.59	70.47 ± 0.85	**73.74 ± 0.88**
Siamese_TPL	53.75 ± 4.16	64.45 ± 1.22	65.83 ± 1.19	67.49 ± 2.19	69.47 ± 1.20	71.15 ± 1.11
Siamese_TPL_A	50.73 ± 1.98	64.80 ± 1.48	67.51 ± 1.41	68.04 ± 1.07	69.97 ± 1.55	71.68 ± 0.52

(b) Micro F1 Score

Table 8: CiteSeer dataset results: (a) Micro Accuracy and (b) Micro F1 across few-shot splits

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	**69.64 ± 0.86**	**73.05 ± 0.42**	77.15 ± 0.22	79.28 ± 0.23	81.01 ± 0.94	81.33 ± 0.31
CSR	69.03 ± 1.24	68.76 ± 0.77	76.55 ± 0.45	77.57 ± 1.48	81.00 ± 0.45	81.52 ± 0.71
ER	67.59 ± 0.95	71.83 ± 1.98	77.04 ± 0.54	79.44 ± 0.31	81.49 ± 0.52	81.58 ± 0.48
DropMessage	67.56 ± 1.40	70.93 ± 2.29	77.42 ± 0.52	**80.23 ± 0.52**	**81.98 ± 0.50**	82.21 ± 0.94
DropNode	67.83 ± 1.45	70.79 ± 2.29	77.45 ± 0.82	79.75 ± 0.51	81.43 ± 0.66	81.70 ± 0.55
DropEdge	68.28 ± 1.48	70.83 ± 1.66	77.00 ± 0.51	79.85 ± 0.43	81.29 ± 0.50	82.05 ± 0.74
DropAttributes	67.48 ± 1.26	69.23 ± 1.37	77.31 ± 0.34	79.85 ± 0.77	81.55 ± 0.68	82.02 ± 0.60
Augmentation	68.26 ± 1.07	70.91 ± 2.00	77.94 ± 0.70	80.19 ± 0.53	81.55 ± 0.65	81.33 ± 0.59
Rewiring	69.09 ± 0.77	71.29 ± 2.12	76.90 ± 0.60	79.96 ± 0.55	81.39 ± 0.48	82.28 ± 0.58
GraphCL	68.25 ± 1.74	71.20 ± 2.38	76.96 ± 0.85	80.12 ± 0.81	81.42 ± 0.55	82.37 ± 0.60
ProtoNet	67.04 ± 1.27	68.91 ± 3.27	71.43 ± 2.68	78.37 ± 1.53	79.72 ± 3.18	81.35 ± 1.13
Siamese_PW	67.14 ± 1.10	70.92 ± 1.13	**77.93 ± 0.49**	80.12 ± 1.00	81.03 ± 0.36	81.96 ± 0.63
Siamese_PW_A	67.93 ± 1.53	70.40 ± 1.01	77.66 ± 0.68	79.94 ± 0.52	80.91 ± 0.70	82.04 ± 0.32
Siamese_TPL	68.86 ± 1.27	72.18 ± 1.37	76.94 ± 0.81	79.85 ± 0.65	81.52 ± 1.09	81.93 ± 0.84
Siamese_TPL_A	69.29 ± 1.24	71.57 ± 1.09	76.91 ± 0.92	79.34 ± 0.64	81.75 ± 0.82	**82.51 ± 0.97**

(a) Micro Accuracy

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	**69.53 ± 0.90**	**72.95 ± 0.46**	77.12 ± 0.21	79.27 ± 0.24	81.07 ± 0.93	81.44 ± 0.33
CSR	68.92 ± 1.22	68.55 ± 0.75	76.55 ± 0.45	77.59 ± 1.51	81.05 ± 0.44	81.55 ± 0.73
ER	67.38 ± 0.97	71.73 ± 2.11	77.03 ± 0.56	79.42 ± 0.34	81.55 ± 0.53	81.65 ± 0.48
DropMessage	67.41 ± 1.40	70.75 ± 2.46	77.41 ± 0.53	**80.22 ± 0.52**	**82.02 ± 0.50**	82.28 ± 0.95
DropNode	67.65 ± 1.41	70.61 ± 2.42	77.44 ± 0.83	79.76 ± 0.48	81.51 ± 0.62	81.79 ± 0.55
DropEdge	68.08 ± 1.51	70.62 ± 1.75	76.98 ± 0.52	79.89 ± 0.42	81.36 ± 0.52	82.13 ± 0.75
DropAttributes	67.23 ± 1.24	68.96 ± 1.43	77.30 ± 0.34	79.84 ± 0.80	81.61 ± 0.68	82.12 ± 0.59
Augmentation	68.12 ± 1.04	70.69 ± 2.16	**77.91 ± 0.70**	80.20 ± 0.55	81.60 ± 0.65	81.42 ± 0.59
Rewiring	68.97 ± 0.79	71.13 ± 2.22	76.87 ± 0.62	79.98 ± 0.53	81.44 ± 0.50	82.35 ± 0.58
GraphCL	68.05 ± 1.92	71.03 ± 2.54	76.95 ± 0.85	80.12 ± 0.81	81.49 ± 0.54	82.45 ± 0.61
ProtoNet	66.72 ± 1.33	68.67 ± 3.49	71.33 ± 2.82	78.34 ± 1.53	79.69 ± 3.40	81.36 ± 1.14
Siamese_PW	67.04 ± 1.02	70.75 ± 1.17	77.93 ± 0.48	80.12 ± 1.01	81.15 ± 0.35	82.04 ± 0.61
Siamese_PW_A	67.78 ± 1.54	70.09 ± 1.13	77.64 ± 0.67	79.93 ± 0.52	81.01 ± 0.70	82.13 ± 0.29
Siamese_TPL	68.71 ± 1.23	72.01 ± 1.56	76.92 ± 0.81	79.87 ± 0.65	81.58 ± 1.08	81.98 ± 0.85
Siamese_TPL_A	69.11 ± 1.29	71.28 ± 1.27	76.87 ± 0.92	79.42 ± 0.65	81.80 ± 0.82	**82.57 ± 0.95**

(b) Micro F1 Score

Table 9: PubMed dataset results: (a) Micro Accuracy and (b) Micro F1 across few-shot splits

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	85.32 ± 0.55	88.14 ± 0.84	88.12 ± 0.78	89.52 ± 0.65	90.09 ± 0.62	90.13 ± 0.73
CSR	**86.85 ± 0.61**	89.14 ± 0.60	**90.56 ± 0.50**	90.40 ± 0.83	91.02 ± 0.47	**91.44 ± 0.52**
ER	85.92 ± 0.79	88.32 ± 0.72	89.12 ± 0.52	90.26 ± 0.61	90.26 ± 0.68	90.77 ± 0.94
DropMessage	86.19 ± 0.75	88.69 ± 0.49	89.13 ± 0.62	90.39 ± 0.25	90.93 ± 0.62	90.92 ± 0.59
DropNode	85.49 ± 0.87	88.43 ± 0.44	88.80 ± 0.70	89.86 ± 0.41	90.53 ± 0.62	90.50 ± 0.69
DropEdge	86.42 ± 0.48	88.37 ± 0.96	88.87 ± 0.75	89.93 ± 0.57	90.37 ± 0.54	90.79 ± 0.39
DropAttributes	85.70 ± 1.99	88.58 ± 0.51	89.00 ± 0.66	90.61 ± 0.45	**91.06 ± 0.66**	90.94 ± 0.79
Augmentation	82.53 ± 1.67	85.48 ± 2.47	85.77 ± 1.11	88.30 ± 0.88	87.99 ± 2.12	87.82 ± 0.96
Rewiring	86.46 ± 0.94	88.69 ± 0.82	89.07 ± 0.70	90.08 ± 0.69	90.26 ± 0.74	90.27 ± 0.80
GraphCL	86.00 ± 0.46	87.82 ± 0.64	88.39 ± 0.60	89.88 ± 0.41	89.98 ± 0.71	90.40 ± 0.59
ProtoNet	77.57 ± 3.26	80.11 ± 3.47	81.92 ± 3.23	79.08 ± 3.78	79.06 ± 4.86	85.55 ± 1.06
Siamese_PW	85.74 ± 2.46	89.08 ± 0.59	89.70 ± 0.59	90.35 ± 0.69	90.99 ± 0.53	91.34 ± 0.39
Siamese_PW_A	85.96 ± 1.83	**89.18 ± 0.62**	89.87 ± 0.62	**90.70 ± 0.40**	90.90 ± 0.42	91.38 ± 0.32
Siamese_TPL	85.68 ± 0.69	88.34 ± 0.49	88.48 ± 1.14	89.80 ± 0.65	90.66 ± 0.48	90.53 ± 0.65
Siamese_TPL_A	86.15 ± 0.91	88.62 ± 0.78	88.77 ± 0.54	89.98 ± 0.66	90.47 ± 0.57	90.47 ± 0.55

(a) Micro Accuracy

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	85.10 ± 0.55	88.07 ± 0.84	88.35 ± 0.74	89.69 ± 0.61	90.23 ± 0.57	90.30 ± 0.73
CSR	**86.57 ± 0.59**	89.03 ± 0.63	**90.49 ± 0.57**	90.37 ± 0.83	90.99 ± 0.50	**91.47 ± 0.53**
ER	85.68 ± 0.85	88.25 ± 0.72	89.33 ± 0.49	90.38 ± 0.60	90.42 ± 0.64	90.89 ± 0.96
DropMessage	86.02 ± 0.74	88.59 ± 0.50	89.26 ± 0.62	90.49 ± 0.28	91.01 ± 0.58	91.02 ± 0.61
DropNode	85.26 ± 0.90	88.43 ± 0.46	89.05 ± 0.70	90.00 ± 0.42	90.69 ± 0.60	90.69 ± 0.70
DropEdge	86.21 ± 0.51	88.33 ± 0.87	89.01 ± 0.70	90.02 ± 0.51	90.50 ± 0.47	90.95 ± 0.40
DropAttributes	85.46 ± 1.93	88.54 ± 0.54	89.15 ± 0.69	90.69 ± 0.43	**91.17 ± 0.63**	91.05 ± 0.78
Augmentation	82.33 ± 1.80	85.53 ± 2.57	86.02 ± 1.09	88.41 ± 0.92	88.09 ± 2.25	88.00 ± 0.95
Rewiring	86.26 ± 1.00	88.68 ± 0.82	89.19 ± 0.57	90.22 ± 0.68	90.37 ± 0.76	90.46 ± 0.79
GraphCL	85.81 ± 0.46	87.76 ± 0.58	88.57 ± 0.62	90.00 ± 0.41	90.15 ± 0.67	90.54 ± 0.63
ProtoNet	78.01 ± 3.13	80.59 ± 3.34	82.34 ± 3.11	79.59 ± 3.76	79.62 ± 4.66	85.87 ± 1.00
Siamese_PW	85.45 ± 2.62	89.01 ± 0.60	89.76 ± 0.62	90.42 ± 0.68	91.06 ± 0.50	91.43 ± 0.37
Siamese_PW_A	85.68 ± 1.94	**89.10 ± 0.65**	89.89 ± 0.65	**90.81 ± 0.33**	90.96 ± 0.44	**91.47 ± 0.32**
Siamese_TPL	85.41 ± 0.69	88.23 ± 0.48	88.57 ± 1.09	89.93 ± 0.61	90.72 ± 0.45	90.65 ± 0.63
Siamese_TPL_A	85.93 ± 0.92	88.47 ± 0.78	88.88 ± 0.50	90.08 ± 0.63	90.52 ± 0.56	90.56 ± 0.56

(b) Micro F1 Score

Table 10: CS dataset results: (a) Micro Accuracy and (b) Micro F1 across few-shot splits

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	91.80 ± 0.62	93.96 ± 0.32	93.94 ± 0.20	94.02 ± 0.26	93.88 ± 0.29	94.04 ± 0.21
CSR	92.18 ± 0.52	**94.78 ± 0.25**	**94.78 ± 0.35**	94.16 ± 0.63	**94.42 ± 0.28**	94.52 ± 0.28
ER	92.32 ± 0.66	94.08 ± 0.43	94.22 ± 0.52	93.94 ± 0.49	93.88 ± 0.21	93.72 ± 0.28
DropMessage	**93.06 ± 0.48**	94.02 ± 0.49	94.14 ± 0.22	94.16 ± 0.15	94.06 ± 0.20	93.82 ± 0.21
DropNode	92.62 ± 0.43	94.24 ± 0.27	94.10 ± 0.56	93.92 ± 0.55	93.80 ± 0.66	93.94 ± 0.16
DropEdge	92.98 ± 0.96	93.68 ± 0.74	94.28 ± 0.44	94.12 ± 0.17	94.06 ± 0.46	94.02 ± 0.24
DropAttributes	91.96 ± 0.81	94.18 ± 0.42	94.30 ± 0.33	93.84 ± 0.30	93.86 ± 0.24	94.02 ± 0.16
Augmentation	92.18 ± 1.40	93.58 ± 0.28	94.00 ± 0.17	93.58 ± 0.70	93.84 ± 0.42	93.16 ± 0.24
Rewiring	92.32 ± 0.49	94.10 ± 0.33	94.18 ± 0.57	94.34 ± 0.30	93.68 ± 0.28	94.00 ± 0.73
ProtoNet	88.10 ± 2.87	90.26 ± 1.34	91.90 ± 0.38	92.32 ± 0.26	92.40 ± 0.66	92.82 ± 0.49
Siamese_PW	92.92 ± 0.82	94.20 ± 0.53	94.60 ± 0.15	94.40 ± 0.30	94.04 ± 0.29	**94.66 ± 0.68**
Siamese_PW_A	92.62 ± 0.65	94.36 ± 0.19	94.48 ± 0.17	**94.52 ± 0.28**	94.30 ± 0.41	94.46 ± 0.39
Siamese_TPL	91.82 ± 0.47	93.90 ± 0.28	93.58 ± 0.33	93.82 ± 0.48	94.02 ± 0.16	93.92 ± 0.23
Siamese_TPL_A	92.50 ± 1.16	94.00 ± 0.32	94.08 ± 0.50	94.32 ± 0.56	94.00 ± 0.36	94.00 ± 0.25

(a) Micro Accuracy

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	91.79 ± 0.62	93.92 ± 0.34	94.03 ± 0.17	94.07 ± 0.25	93.93 ± 0.29	94.08 ± 0.19
CSR	92.21 ± 0.50	**94.80 ± 0.23**	**94.78 ± 0.35**	94.23 ± 0.60	**94.45 ± 0.26**	94.54 ± 0.26
ER	92.30 ± 0.62	94.06 ± 0.44	94.23 ± 0.46	94.01 ± 0.43	93.93 ± 0.20	93.79 ± 0.27
DropMessage	93.00 ± 0.49	94.01 ± 0.47	94.20 ± 0.19	94.21 ± 0.14	94.10 ± 0.20	93.86 ± 0.21
DropNode	92.59 ± 0.46	94.22 ± 0.28	94.08 ± 0.57	93.98 ± 0.53	93.88 ± 0.62	93.99 ± 0.16
DropEdge	**93.02 ± 0.92**	93.69 ± 0.73	94.27 ± 0.40	94.15 ± 0.13	94.10 ± 0.44	94.07 ± 0.23
DropAttributes	91.96 ± 0.81	94.18 ± 0.39	94.33 ± 0.29	93.92 ± 0.29	93.93 ± 0.23	94.07 ± 0.15
Augmentation	92.25 ± 1.30	93.62 ± 0.31	94.05 ± 0.18	93.67 ± 0.66	93.91 ± 0.40	93.28 ± 0.20
Rewiring	92.34 ± 0.53	94.09 ± 0.31	94.18 ± 0.48	94.37 ± 0.30	93.70 ± 0.27	94.05 ± 0.68
ProtoNet	88.56 ± 2.65	90.54 ± 1.30	92.15 ± 0.35	92.52 ± 0.26	92.62 ± 0.61	92.99 ± 0.45
Siamese_PW	92.87 ± 0.81	94.21 ± 0.50	94.61 ± 0.17	94.43 ± 0.29	94.08 ± 0.29	**94.67 ± 0.66**
Siamese_PW_A	92.66 ± 0.62	94.36 ± 0.18	94.45 ± 0.16	**94.54 ± 0.26**	94.31 ± 0.41	94.48 ± 0.38
Siamese_TPL	91.89 ± 0.53	93.91 ± 0.31	93.69 ± 0.31	93.90 ± 0.45	94.06 ± 0.16	93.98 ± 0.20
Siamese_TPL_A	92.56 ± 1.11	93.99 ± 0.32	94.15 ± 0.42	94.36 ± 0.52	94.02 ± 0.36	94.06 ± 0.24

(b) Micro F1 Score

Table 11: Physics dataset results: (a) Micro Accuracy and (b) Micro F1 across few-shot splits

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	63.96 ± 2.61	71.46 ± 1.66	71.62 ± 1.64	77.14 ± 2.58	75.62 ± 2.36	74.62 ± 2.57
CSR	**69.90 ± 2.50**	**75.58 ± 1.27**	**76.90 ± 2.22**	76.68 ± 2.01	74.64 ± 1.60	76.00 ± 1.40
ER	62.54 ± 3.04	68.90 ± 2.48	73.10 ± 2.63	74.92 ± 3.91	70.50 ± 4.70	70.28 ± 1.43
DropMessage	64.58 ± 0.99	64.16 ± 15.86	58.14 ± 17.84	75.96 ± 2.37	71.42 ± 1.23	72.82 ± 3.17
DropNode	63.46 ± 3.08	73.66 ± 0.59	71.08 ± 1.16	76.00 ± 2.06	73.12 ± 3.04	72.76 ± 2.72
DropEdge	64.30 ± 1.40	74.26 ± 2.25	71.40 ± 1.75	75.34 ± 3.64	**77.84 ± 0.80**	73.32 ± 4.34
DropAttributes	65.96 ± 1.11	72.74 ± 2.40	72.40 ± 1.41	74.90 ± 4.19	73.50 ± 2.55	74.76 ± 3.46
Augmentation	36.68 ± 3.12	31.16 ± 3.21	39.48 ± 2.00	37.70 ± 1.24	40.06 ± 3.67	36.78 ± 1.52
Rewiring	53.24 ± 13.82	69.18 ± 1.75	69.52 ± 3.15	74.50 ± 2.33	70.52 ± 2.84	72.30 ± 2.47
GraphCL	62.44 ± 5.10	62.94 ± 13.88	69.54 ± 1.70	74.16 ± 3.85	72.98 ± 4.81	75.82 ± 2.10
ProtoNet	25.34 ± 3.68	49.16 ± 4.12	46.42 ± 9.83	44.72 ± 7.11	53.06 ± 11.92	50.48 ± 6.07
Siamese_PW	67.86 ± 1.16	70.68 ± 1.96	74.82 ± 2.22	**77.22 ± 2.59**	67.88 ± 13.52	**78.24 ± 1.28**
Siamese_PW_A	66.20 ± 3.26	74.54 ± 1.76	74.22 ± 2.87	74.66 ± 2.65	76.14 ± 0.81	76.00 ± 2.84
Siamese_TPL	60.48 ± 1.21	73.08 ± 1.79	73.60 ± 2.19	77.20 ± 3.24	76.88 ± 1.11	77.64 ± 3.01
Siamese_TPL_A	59.48 ± 1.28	71.22 ± 2.51	73.54 ± 2.01	74.18 ± 2.07	73.54 ± 2.06	73.18 ± 0.40

(a) Micro Accuracy

Method	FS5	FS10	FS20	FS40	FS60	FS80
Baseline	66.29 ± 2.32	72.80 ± 1.61	72.19 ± 1.51	77.49 ± 2.34	76.40 ± 2.55	75.11 ± 2.55
CSR	**69.76 ± 2.01**	**76.03 ± 1.41**	**76.70 ± 2.26**	76.89 ± 2.20	74.91 ± 1.49	76.27 ± 1.46
ER	64.27 ± 2.92	70.15 ± 2.76	73.63 ± 2.63	75.43 ± 3.68	71.25 ± 4.52	70.97 ± 1.38
DropMessage	66.03 ± 0.70	62.99 ± 20.79	51.49 ± 26.22	76.26 ± 2.17	71.94 ± 1.29	73.45 ± 2.94
DropNode	64.68 ± 2.71	74.71 ± 0.69	71.54 ± 1.23	76.34 ± 1.95	73.61 ± 2.79	73.29 ± 2.63
DropEdge	65.24 ± 1.59	75.35 ± 2.23	71.81 ± 1.56	75.67 ± 3.44	**78.05 ± 0.69**	73.93 ± 4.28
DropAttributes	67.03 ± 0.77	73.95 ± 2.29	72.96 ± 1.48	75.34 ± 3.89	74.01 ± 2.42	75.30 ± 3.40
Augmentation	27.14 ± 8.19	28.68 ± 1.39	35.24 ± 2.39	33.39 ± 1.32	36.09 ± 3.12	30.60 ± 5.89
Rewiring	53.42 ± 7.42	71.14 ± 1.33	70.39 ± 2.93	75.07 ± 2.34	70.80 ± 3.17	72.86 ± 2.59
GraphCL	63.76 ± 4.54	60.68 ± 4.87	70.06 ± 1.79	74.66 ± 3.71	73.57 ± 4.76	76.16 ± 2.11
ProtoNet	25.87 ± 4.84	51.62 ± 4.60	47.12 ± 12.19	46.60 ± 7.21	54.14 ± 12.02	53.00 ± 6.24
Siamese_PW	67.90 ± 1.00	71.61 ± 2.15	75.33 ± 2.37	77.66 ± 2.49	67.96 ± 14.01	**78.51 ± 1.39**
Siamese_PW_A	67.24 ± 2.68	75.25 ± 1.60	74.91 ± 2.96	75.03 ± 2.51	76.44 ± 0.77	76.64 ± 2.90
Siamese_TPL	62.64 ± 1.02	74.42 ± 1.50	74.36 ± 2.22	**77.69 ± 3.06**	77.28 ± 0.96	78.16 ± 2.91
Siamese_TPL_A	62.00 ± 0.91	72.82 ± 2.17	74.48 ± 2.23	74.74 ± 1.95	74.09 ± 1.94	73.72 ± 0.34

(b) Micro F1 Score

Table 12: Amazon dataset results: (a) Micro Accuracy and (b) Micro F1 across few-shot splits