LEDA: Latent Semantic Distribution Alignment for Multi-domain Graph Pre-training

Lianze Shan[†]
shanlz2119@tju.edu.cn
Tianjin University
Tianjin, China

Jitao Zhao[†]
zjtao@tju.edu.cn
Tianjin University
Tianjin, China

Dongxiao He[*]
hedongxiao@tju.edu.cn
Tianjin University
Tianjin, China

Siqi Liu
siqiliu@tju.edu.cn
Tianjin University
Tianjin, China

Jiaxu Cui
cjx@jlu.edu.cn
Jilin University
Changchun, China

Weixiong Zhang
weixiong.zhang@polyu.edu.hk
The Hong Kong Polytechnic University
Kowloon, Hong Kong

Abstract

Recent advances in generic large models, such as GPT-xyz and DeepSeek, have motivated the introduction of universality to graph pre-training, aiming to learn rich and generalizable knowledge across diverse domains using graph representations to improve performance in various downstream applications. However, most existing methods face challenges in learning effective knowledge from generic graphs, primarily due to simplistic data alignment and limited training guidance. The issue of simplistic data alignment arises from the use of a straightforward unification for highly diverse graph data, which fails to align semantics and misleads pre-training models. The problem with limited training guidance lies in the arbitrary application of in-domain pre-training paradigms to cross-domain scenarios. While it is effective in enhancing discriminative representation in one data space, it struggles to capture effective knowledge from many graphs. To address these challenges, we propose a novel Latent sEmantic Distribution Alignment (LEDA) model for universal graph pre-training. Specifically, we first introduce a dimension projection unit to adaptively align diverse domain features into a shared semantic space with minimal information loss. Furthermore, we design a variational semantic inference module to obtain the shared latent distribution. The distribution is then adopted to guide the domain projection, aligning it with shared semantics across domains and ensuring cross-domain semantic learning. LEDA exhibits strong performance across a broad range of graphs and downstream tasks. Remarkably, in few-shot cross-domain settings, it significantly outperforms in-domain baselines and advanced universal pre-training models.

[†]Both authors contributed equally to this research.
[*]Corresponding author.

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

CCS Concepts

• **Computing methodologies** → **Neural networks**.

Keywords

Graph Neural Networks, Graph Self-Supervised Learning, Graph Pre-training

ACM Reference Format:
Lianze Shan, Jitao Zhao, Dongxiao He, Siqi Liu, Jiaxu Cui, and Weixiong Zhang. 2018. LEDA: Latent Semantic Distribution Alignment for Multi-domain Graph Pre-training. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Graph pre-training has emerged as a powerful framework that aims to design proxy tasks and train encoders to learn knowledge with high generalizability [41]. It has been widely applied in various label-scarce scenarios, such as social networks [43], recommendation systems [12], and molecular analysis [38], and has played a pivotal role in graph learning research and development. Recently, it has served as a core technique for developing graph foundation models [15, 45, 46], enabling them to acquire universal knowledge that enhances performance across diverse downstream tasks.

Recent advances in graph pre-training have shifted from specific, application-oriented, self-supervised paradigms [35, 37, 53] to generalizable and universal pre-training frameworks [22, 29, 45]. Classical graph self-supervised pre-training frameworks train encoders to learn graph representations by designing contrastive or generative proxy tasks [13, 16, 35, 53]. While effective on applications within the same domains, these frameworks are difficult to generalize across diverse graphs due to the inherent high disparities of graph data [50]. With the advancement of universal large models [1, 4], universal graph pre-training models have drawn much attention lately, aiming to learn more generalizable knowledge by training on a variety of graphs, enabling them to be readily applicable to many downstream tasks [22, 50, 52]. To achieve this objective, early methods depended on structural encoding to ensure the generality of attribute-free graphs [29]. Therefore, these methods are restricted to structural generalizability. More recent methods employ Language Models (LM) by converting node attributes to textual

descriptions, enabling semantic encoding and cross-graph attribute alignment [22]. Another paradigm leverages Matrix Factorization (MF) methods, such as Principal Component Analysis (PCA) and Singular Value Decomposition (SVD), to project node features into a lower-dimensional space for alignment [46, 50]. Based on the aligned data, most existing methods explore contrastive learning or link reconstruction objectives for model training [22, 29, 46, 50].

However, these methods face challenges in learning effective knowledge from numerous general graphs due to their simplistic data alignment strategies and limited knowledge acquisition capabilities. Firstly, simplistic alignment strategies lack the flexibility and expressiveness to handle highly diverse graphs, resulting in sub-optimal alignment when applied to generic graphs. Specifically, structure-based methods fall short on attributed graphs as they ignore node feature semantics. LM-based methods rely on textual attributes and are therefore inadequate for handling graphs with non-textual features. MF-based methods align features by identifying principal variance directions but may retain noise or lose task-relevant semantics. Moreover, the projected dimensions lack semantic clarity, which can potentially lead to confusion during pre-training. Secondly, limited knowledge learning stems from the arbitrary application of in-domain pre-training paradigms to multi-domain scenarios. Although the widely used contrastive learning and link reconstruction perform well in in-domain scenarios, they may become sub-optimal in multi-domain settings, as semantic ambiguity across samples makes it difficult to establish meaningful relations in the unified space. We further explore this issue in detail in Section 4.2. These limitations motivated us to consider *how to design alignment strategies that preserve semantics across diverse graphs* and *how to defince pre-training objectives that learn effective knowledge from multi-domain graphs.*

To address these challenges, we propose a novel **L**atent s**E**mantic **D**istribution **A**lignment (LEDA) model for multi-domain universal graph pre-training. Specifically, we introduced a learnable domain projection unit that captures projection directions aligned with shared semantics, enabling the unification of multi-domain graphs. Instead of relying on arbitrary variance-maximizing projections, we maximize the mutual information between original features and their projected representations to ensure each dimension retains essential information from the original feature. Building upon the aligned feature, we further incorporate a latent distribution alignment module, which explicitly aligns the posterior distributions of individual domains with a shared prior. This process also provides semantic guidance for the domain projection unit, promoting alignment along consistent directions across different domains. We make the following contributions:

- We revealed that existing graph pre-training methods are incapable of learning generalizable knowledge from multi-domain graphs due to their simplistic data alignment strategies and limited knowledge learning. We adopted a variational semantic modeling perspective to overcome these challenges.
- We proposed a novel universal graph pre-training approach, which integrates a domain projection unit for adaptive feature projection and a latent distribution alignment module to jointly optimize latent inference and generalizable semantic modeling.
- We conducted extensive experiments to assess the effectiveness of our model, including cross-domain node classification, cross-domain graph classification, and few-shot scenarios using seven widely-used node classification datasets and four representative graph classification benchmarks.

2 Related Work

Graph Representation Learning. Graphs are ubiquitous in real-world scenarios, such as social networks [43], bioinformatics [47] and recommendation systems [12]. This leads to an extensive focus on Graph Representation Learning (GRL), which aims to encode graph data into low-dimensional continuous embedding space for every step in downstream tasks [3]. Early works rely on shallow embedding methods, including matrix factorization [48] and random walk-based [18] methods, which struggle to jointly encode node features and topology. In recent years, Graph Neural Networks (GNNs) have achieved significant success, as demonstrated by models such as GCN [17], GAT [36] and GraphSAGE [9], which effectively combine node features and topology simultaneously through message passing and aggregation mechanisms. These methods have achieved great success in a wide range of downstream tasks, including node classification, link prediction, and graph classification. However, training effective GNNs requires large amounts of manually labeled data, which is extensive and time-consuming to obtain.

Graph Pre-training. To overcome the limitations of GNNs in label-scarce scenarios, graph pre-training has emerged as an effective paradigm for learning general knowledge and latent graph patterns [41]. As an early paradigm of graph pre-training, Graph Self-Supervised Learning (GSSL) has been extensively studied [10, 11, 14]. GSSL mines data itself to generate self-supervised samples and designs proxy tasks for training GNN encoders in a self-supervised way. Recent advances in GSSL have predominantly evolved along two directions: (i) contrastive-based methods [35, 37, 53], which aim to maximize mutual information between different views of original graph, and (ii) generative-based methods [13, 16, 19], which focus on reconstructing masked graph signals. While these methods have achieved remarkable success in label-scarce scenarios, most of them remain confined to in-domain distribution, making it difficult to extract transferable knowledge across diverse graph distributions.

Inspired by the success of unified pre-training large models [4], recent efforts have focused on developing unified graph pre-training models. Given the reliance of GNNs on fixed input dimensions and the semantic inconsistency of node features across graphs, some methods [6, 29] discard raw node features and instead encode topology-based positional information derived purely from graph structure. While these methods enable better generalization across attribute-free graphs, their inability to incorporate attribute information results in suboptimal performance on attributed graphs, which are common in real-world scenarios. To unify attribute information across graphs, methods such as OFA [22] and ZEROG [20] convert graphs into text and utilize Language Models (LMs) to re-encode node attributes. Building on the unified input, OFA [22] introduces virtual nodes to integrate essential information

derived from the original graph and LMs, which is then propagated via GNNs. ZEROG [20] proposes a prompt-based subgraph sampling mechanism that captures semantic relevance through selected prompt nodes, while preserving structural characteristics via local neighborhood aggregation. However, these methods are limited by the assumption that node attributes can be effectively textualized, and the use of LMs incurs substantial computational and resource costs [50]. More recent studies have explored input unification across general graphs by applying dimension reduction techniques such as Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) to align node feature dimensions. FUG [50] revisits the theoretical connection between PCA and contrastive learning, introducing a dimensional encoder to achieve lossless feature alignment. While effective in unifying input dimensions, it heavily relies on data sampling and struggles to capture shared semantics across domains. MDGPT [46] aligns diverse feature dimensions by SVD and introduces the concept of domain tokens to unify semantics from multi-domain graphs. However, the number of domain tokens increases with the number of input domains, which limits the model's flexibility and scalability.

Despite the progress made in recent efforts on universal graph pre-training, existing approaches still face intrinsic challenges in multi-domain settings. These limitations motivate us to reconsider how to design a universal graph pre-training framework that can effectively learn transferable knowledge from multi-domain graphs in a scalable and generalizable manner.

3 Notations and Preliminary

We now present the preliminary concepts and notations used in the paper. Sets are denoted by calligraphic letters (e.g., $\mathcal{G}$), matrices are represented in bold capital letters (e.g., $\mathbf{X}$), vectors are expressed in bold lowercase letters (e.g., $\mathbf{x}$, which denotes the vector of matrix $\mathbf{X}$), and scalars are denoted by lowercase letters (e.g., x).

Graph data. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we define $\mathcal{V}$ as the node set, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the edge set. Each node v_i is associated with a feature vector $\mathbf{x}_i$, and the collective features of all nodes in graph $\mathcal{G}$ form the feature matrix matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n is the number of nodes and d is the feature dimension. The topological structure of $\mathcal{G}$ is specified by its adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$, where the entry a_{ij} is defined as $a_{ij} = 1$ if $e_{ij} \in \mathcal{E}$ or $a_{ij} = 0$ otherwise.

Graph Neural Networks (GNNs). Graph pre-training involves training a universal encoder that can extract generalizable knowledge from diverse graphs that can be rapidly adapted and transferred to a variety of downstream tasks. In most graph pre-training methods, GNNs [40, 51] are used as the backbone encoder, which encode node representations through the message passing and aggregation framework:

$$\mathbf{h}_i^{(l+1)} = \text{UPDATE}^{(l)}\left(\mathbf{h}_i^{(l)}, \text{AGGREGATE}^{(l)}\left(\{\mathbf{h}_j^{(l)} : j \in \mathcal{N}(i)\}\right)\right), \tag{1}$$

where $\mathbf{h}_i^{(l)}$ denotes the l-layer representation of node v_i, $\mathcal{N}(i)$ denotes the set of neighbors of v_i, $\text{AGGREGATE}^{(l)}$ and $\text{UPDATE}^{(l)}$ are layer-specific functions, and the primary differences among various GNNs lie in how these two functions are designed. In our work, we consider the Graph Convolutional Network (GCN) [17] as the backbone encoder for graph pre-training, which defines the aggregation as a normalized sum over neighbors:

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

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops, and $\tilde{\mathbf{D}}$ is the corresponding diagonal degree matrix, $\mathbf{W}^l$ is a learnable weight matrix and σ is a non-linear activation function.

Multi-domain Graph Pre-training. Training the universal GNN encoder mentioned above requires pre-training on multi-domain graphs to learn generalizable knowledge. And we define the collection of multi-domain graphs as $\mathcal{D}_{\mathcal{G}} = \{\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(M)}\}$, where each $\mathcal{G}^{(k)} = (\mathcal{V}^{(k)}, \mathcal{E}^{(k)})$ denotes the k-th graph instance in the set, and M is the number of graphs. These graphs often exhibit distinct characteristics, particularly in node feature dimensions and semantics. This poses challenges for training a universal GNN encoder due to the fixed input requirement of $\mathbf{W}^l$. To address this, some methods apply data pre-processing techniques such as singular value decomposition to align features:

$$\mathbf{X}^i \approx \mathbf{U}_k^i \Sigma_k^i \mathbf{V}_k^{i^\top}, \mathbf{X}_{\text{align}}^i = \mathbf{X}^i \mathbf{V}_k^i, \tag{3}$$

where $\mathbf{U}_k^i \in \mathbb{R}^{n \times k}$, $\Sigma_k^i \in \mathbb{R}^{k \times k}$, and $\mathbf{V}_k^i \in \mathbb{R}^{d \times k}$ are the top-k singular vectors and values. The reduced representation $\mathbf{X}_{\text{align}}^i$ is then used as the aligned node features.

4 Methodology

In this section, we introduce our **L**atent s**E**mantic **D**istribution **A**lignment (LEDA) approach for multi-domain graph pre-training, as illustrated in Figure 1. The goal of LEDA is to learn generalizable distribution from multi-domain graphs, enabling the encoder directly applied to various downstream tasks. We first design a **D**omain **P**rojection **U**nit (DPU) to provide a semantically aligned basis that maps multi-domain features into a unified embedding space while preserving their information through mutual information maximization(in Section 4.1). We then conduct a detailed analysis of the limitation of existing graph pre-training methods in multi-domain scenarios (in Section 4.2). Built upon this, we introduce a **L**atent **D**istribution **A**lignment module (LDA), which learns a shared latent distribution across domains by aligning their posterior distribution to a common prior (in Section 4.3). In addition, a detailed discussion of the assumptions used in this section is provided in Appendix B.

4.1 Domain Projection Unit

A key challenge in multi-domain graph pre-training is the semantic heterogeneity of node features: domains may contain different feature dimensions and semantics, making direct comparison or joint learning ineffective. While dimensionality reduction can unify feature dimensions, it ignores two critical requirements in multi-domain settings: (i) enabling cross-domain semantic alignment while retaining essential semantics, and (ii) avoiding domain semantic conflicts. More discussion about domain semantic conflicts is in Appendix B. To address these challenges, we propose a novel **D**omain **P**rojection **U**nit (DPU), a learnable projection module that maps multi-domain features into a shared semantic subspace while satisfying both requirements through $\mathcal{L}_{\text{align}}$.

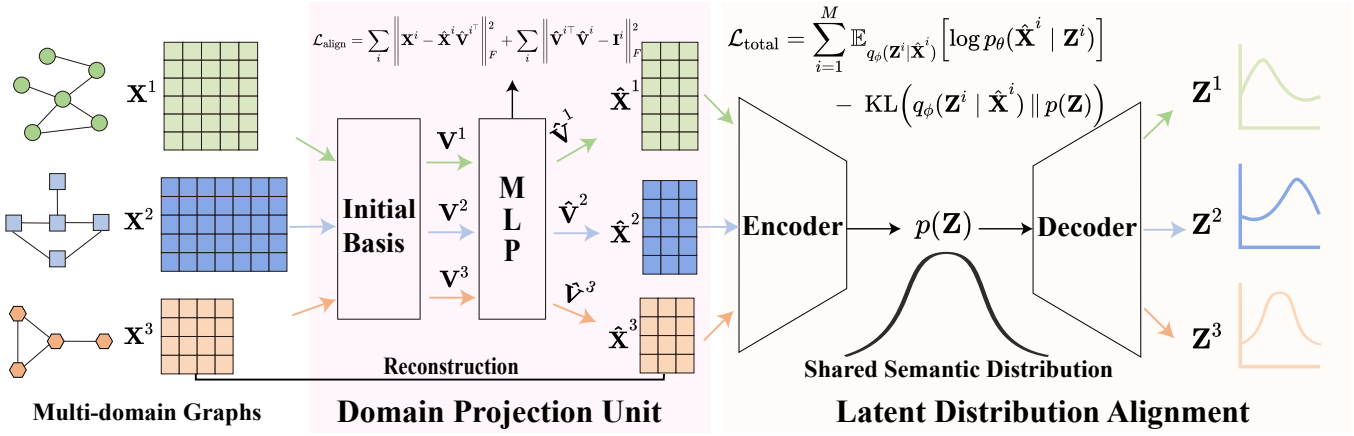

Figure 1: Overview of the proposed LEDA. Given the train dataset $\mathcal{T}_{\mathcal{G}} = \{\mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^t\}$, we first get their initial projection basis set $\mathcal{P}_{\mathcal{G}} = \{\mathbf{V}^1, \mathbf{V}^2, \ldots, \mathbf{V}^t\}$ by SVD. Subsequently, this set of projection basis is processed by a learnable multi-layer perceptron, which is optimized by $\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{recon}} + \lambda \cdot \mathcal{L}_{\text{ortho}}$. Furthermore, we encode the unified feature processed by DPU through a single-layer GCN and align the posterior distribution with a shared latent distribution. Finally, we jointly optimize the parameters of DPU and LDA using the loss function $\mathcal{L}_{\text{total}}$.

Specifically, for each graph $\mathcal{G}^i = (\mathbf{X}^i, \mathbf{A}^i)$, we first compute an initial projection basis $\mathbf{V}^i \in \mathbb{R}^{d_i \times k}$ via SVD to capture the dominant variance in $\mathbf{X}^i$. This provides a stable, low-rank initialization that avoids the instability of random projection in high dimensions. We then refine $\mathbf{V}^i$ using a domain-shared learnable transformation function $\text{Trans}(\cdot)$, implemented as a two-layer MLP:

$$\hat{\mathbf{V}}^i = \text{Trans}(\mathbf{V}^i) = \text{ReLU}(\mathbf{V}^i\mathbf{W}_1 + \mathbf{b}_1) \cdot \mathbf{W}_2 + \mathbf{b}_2, \tag{4}$$

where the same parameters $\{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2\}$ are applied to all domains. This shared-parameter forces all domains to adapt their basis into a unified semantic space, laying the foundation for cross-domain compatibility. The aligned node representations are obtained as $\hat{\mathbf{X}}^i = \mathbf{X}^i\hat{\mathbf{V}}^i$, where $\hat{\mathbf{X}}^i \in \mathbb{R}^{n_i \times m}$. To ensure that this projection retains essential semantics, we introduce a reconstruction loss that encourages $\hat{\mathbf{X}}^i$ to be sufficient for recovering $\mathbf{X}^i$:

$$\mathcal{L}_{\text{recon}} = \sum_i \left\| \mathbf{X}^i - \hat{\mathbf{X}}^i\hat{\mathbf{V}}^{i\top} \right\|_F^2 = \sum_i \left\| \mathbf{X}^i - \mathbf{X}^i\hat{\mathbf{V}}^i\hat{\mathbf{V}}^{i\top} \right\|_F^2, \tag{5}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

However, reconstruction and shared parameters alone are insufficient to resolve the semantic conflicts in the unified embedding space. To address this, we enhance the discriminability of the aligned representations by maximizing the mutual information between the original features $\mathbf{X}^i$ of each domain and their aligned representations $\hat{\mathbf{X}}^i$ as follows:

$$I(\mathbf{X}^i; \hat{\mathbf{X}}^i) = H(\hat{\mathbf{X}}^i) - H(\hat{\mathbf{X}}^i \mid \mathbf{X}^i), \tag{6}$$

where $H(\hat{\mathbf{X}}^i)$ denotes the entropy of the projected features, and $H(\hat{\mathbf{X}}^i \mid \mathbf{X}^i)$ represents the conditional entropy given the original features. Given that the projected representation $\hat{\mathbf{X}}^i$ is deterministically computed from the original features $\mathbf{X}^i$ (i.e., $\hat{\mathbf{X}}^i = \mathbf{X}^i \cdot f(\mathbf{V}^i)$), where $f(\cdot)$ is a deterministic function and $\mathbf{V}^i$ is derived from $\mathbf{X}^i$, the conditional entropy $H(\hat{\mathbf{X}}^i \mid \mathbf{X}^i)$ becomes zero. Therefore, to maximize the mutual information, we maximize the entropy of the projected representation $\hat{\mathbf{X}}^i = \mathbf{X}^i \cdot f(\mathbf{V}^i)$. Since $\mathbf{X}^i$ is deterministic for each domain $\mathcal{G}^i$, this is achieved by encouraging higher entropy in $\hat{\mathbf{V}}^i$. Assuming $\hat{\mathbf{V}}^i$ follows an approximately multivariate Gaussian distribution, the entropy admits the following expression:

$$h(\hat{\mathbf{V}}^i) = \frac{1}{2}\log\left((2\pi e)^m \cdot \det(\Sigma_{\hat{V}^i})\right), \tag{7}$$

where $\Sigma_{\hat{V}^i}$ is the covariance matrix and $\det(\cdot)$ denotes the determinant of the covariance matrix. According to the *Hadamard inequality*, the determinant of $\Sigma_{\hat{V}^i}$ is maximized when the columns of $\hat{\mathbf{V}}^i$ are orthogonal. Therefore, enforcing orthogonality among projection directions increases $h(\hat{\mathbf{V}}^i)$, thus indirectly improving the mutual information $I(\mathbf{X}^i; \hat{\mathbf{X}}^i)$ and enhancing the semantic quality of the representations. We encode this constraint using the following regularization term:

$$\mathcal{L}_{\text{ortho}} = \sum_i \left\| \hat{\mathbf{V}}^{i\top}\hat{\mathbf{V}}^i - \mathbf{I}^i \right\|_F^2, \tag{8}$$

and the final alignment loss is defined as:

$$\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{recon}} + \lambda \cdot \mathcal{L}_{\text{ortho}}, \tag{9}$$

where λ is a hyperparameter that balances information preservation and structural regularization. This joint objective ensures that DPU learns a unified embedding space where features can support cross-domain semantic consistency.

4.2 Analysis of graph pre-training

Existing graph pre-training methods, such as contrastive learning and link prediction [13, 35, 37, 53], are primarily designed for single-domain settings. When applied to multi-domain graphs, they often fail to capture shared semantics across domains due to fundamental misalignments in their optimization objectives.

Limitations of Contrastive Learning. In contrastive learning frameworks (e.g., InfoNCE [26]), the goal is to maximize the similarity between positive node pairs (same domain) while minimizing

it for negative pairs (different domains). This is formalized as:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N}\sum_{i=1}^{N}\log\Bigg(\frac{\sum_{(v_i,v_j)\in\text{Pos}}\exp(\text{sim}(\mathbf{z}_i,\mathbf{z}_j^+)/\tau)}{\sum_{(v_i,v_j)\in\text{Pos}}\exp(\text{sim}(\mathbf{z}_i,\mathbf{z}_j^+)/\tau)+\sum_{(v_i,v_j)\in\text{Neg}}\exp(\text{sim}(\mathbf{z}_i,\mathbf{z}_j^-)/\tau)}\Bigg), \tag{10}$$

where N is the total number of nodes across all domains, $\mathbf{z} = \text{GNN}(\mathbf{A},\mathbf{X})$, $(v_i,v_j) \in \text{Pos}$ when v_i and v_j come from the same domain, $(v_i,v_j) \in \text{Neg}$ when v_i and v_j come from different domains, $\text{sim}(\cdot)$ is the cosine similarity function, and τ is the temperature coefficient. While effective within a single domain, this objective becomes problematic in multi-domain settings. Inspired by [28], we analyze the mutual information across domains to quantify cross-domain alignment.

Definition 1. *(Mutual Information across Domains) Let $\mathcal{D}_{\mathcal{G}^i}$ and $\mathcal{D}_{\mathcal{G}^j}$ denote the data distributions from two different domains $\mathcal{G}^i$ and $\mathcal{G}^j$. The mutual information across domains is defined as:*

$$I(\mathcal{D}_{\mathcal{G}^i};\mathcal{D}_{\mathcal{G}^j}) = \mathbb{E}_{x\sim\mathcal{D}_{\mathcal{G}^i},x'\sim\mathcal{D}_{\mathcal{G}^j}}\left[\log\frac{p(x,x')}{p(x)p(x')}\right], \tag{11}$$

where $p(x,x')$ is the joint distribution over samples from different domains, and $p(x), p(x')$ are marginal distributions.

In contrastive learning, the joint distribution $p(x,x')$ is typically modeled via a similarity-based scoring function $s_{ij} = f(x,x')$, normalized by a constant $Z : p(x,x') = \frac{e^{s_{ij}}}{Z}$. And the marginal distribution is modeld by $p(x)p(x') = C\cdot e^{\xi(x,x')}$, where C is a constant and $e^{\xi(x,x')}$ is a bias term reflecting the difference between the assumption $p(x)p(x') = C$ and the true distribution. Based on this assumption, we give the following proposition:

Proposition 1. *Let the joint distribution over samples from two domains $\mathcal{D}_{\mathcal{G}^i}$ and $\mathcal{D}_{\mathcal{G}^j}$ be modeled as $p(x,x') = e^{s_{ij}}/Z$, where $s_{ij} = f(x,x')$ is a similarity score and $Z = \sum_{x,x'} e^{s_{ij}}$ (or $\int e^{s_{ij}}dxdx'$ in continuous case) is the normalization constant. Further assume the product of marginals satisfies $p(x)p(x') = e^{\xi(x,x')}$. Then the mutual information between the two domains satisfies:*

$$I(\mathcal{D}_{\mathcal{G}^i};\mathcal{D}_{\mathcal{G}^j}) = \mathbb{E}[s_{ij}] - \log Z - \Delta, \tag{12}$$

where $\Delta = \mathbb{E}[\xi(x,x')]$.

Proof. By definition, the mutual information is:

$$I(\mathcal{D}_{\mathcal{G}^i};\mathcal{D}_{\mathcal{G}^j}) = \mathbb{E}_{x,x'}\left[\log\frac{p(x,x')}{p(x)p(x')}\right]. \tag{13}$$

Substituting the assumed forms $p(x,x') = e^{s_{ij}}/Z$ and $p(x)p(x') = e^{\xi(x,x')}$ yields:

$$I(\mathcal{D}_{\mathcal{G}^i};\mathcal{D}_{\mathcal{G}^j}) = \mathbb{E}_{x,x'}\left[\log\left(\frac{e^{s_{ij}}/Z}{e^{\xi(x,x')}}\right)\right]. \tag{14}$$

Simplifying the logarithm gives:

$$I(\mathcal{D}_{\mathcal{G}^i};\mathcal{D}_{\mathcal{G}^j}) = \mathbb{E}_{x,x'}\left[s_{ij} - \log Z - \xi(x,x')\right]. \tag{15}$$

Applying the linearity of expectation separates the terms:

$$I(\mathcal{D}_{\mathcal{G}^i};\mathcal{D}_{\mathcal{G}^j}) = \mathbb{E}[s_{ij}] - \log Z - \mathbb{E}[\xi(x,x')]. \tag{16}$$

Denoting $\Delta = \mathbb{E}[\xi(x,x')]$, we obtain:

$$I(\mathcal{D}_{\mathcal{G}^i};\mathcal{D}_{\mathcal{G}^j}) = \mathbb{E}[s_{ij}] - \log Z - \Delta. \tag{17}$$

□

Proposition 1 reveals a critical limitation of contrastive learning in multi-domain settings: the cross-domain mutual information is directly governed by the expected similarity score $\mathbb{E}[s_{ij}]$. In practice, contrastive objectives explicitly minimize s_{ij} for cross-domain (negative) pairs to enforce domain separation. As a result, $\mathbb{E}[s_{ij}]$ decreases, leading to a reduction in $I(\mathcal{D}_{\mathcal{G}^i};\mathcal{D}_{\mathcal{G}^j})$. While this enhances intra-domain discrimination, it actively suppresses the learning of shared semantics across domains.

Limitations of Link Prediction. Link prediction-based pre-training optimizes a local reconstruction objective: it encourages the model to predict observed edges by maximizing the similarity of connected node pairs. While effective for capturing domain-specific structural patterns, this approach is inherently limited in multi-domain settings for two key reasons. First, it overfits to local topology, ignoring higher-order semantic relationships that are often shared across domains (e.g., functional roles or community memberships). Second, and more critically, it lacks any explicit mechanism for cross-domain alignment: each domain is trained in isolation, so semantically equivalent nodes (e.g., "influential users" in social networks and "highly cited papers" in citation graphs) may be mapped to distant regions in the embedding space. Consequently, link prediction learns representations that are highly specialized to single domain structures but fail to generalize to unseen domains or tasks requiring cross-domain knowledge.

4.3 Latent Distribution Alignment

Motivation of latent distribution alignment. Existing graph pre-training methods often fail to explicitly capture shared semantics across domains, as they either focus on structural similarity or rely on domain-specific objectives. To address this, we propose to learn a shared latent semantic distribution $p(\mathbf{Z})$ from multi-domain graphs, under the assumption that all observed graphs $\{\mathcal{G}^1,\ldots,\mathcal{G}^g\}$ are generated from this common prior [21], with domain-specific variations captured by conditional decoders $p_\theta(\hat{\mathbf{X}}^i \mid \mathbf{Z}^i)$. Based on this assumption, our Latent Distribution Alignment (LDA) module aligns the posterior distribution $q_\phi(\mathbf{Z}^i \mid \hat{\mathbf{X}}^i)$, which is encoded by a shared GCN encoder, to the shared prior $p(\mathbf{Z})$ by minimizing the KL divergence. Because the encoder is shared and the inputs have already been aligned by the DPU, this process enforces that semantically similar samples from distinct domains share a common latent representation. As a result, LDA promotes domain-invariant semantics. In contrast, contrastive learning tends to arbitrarily push cross-domain pairs apart by treating them as negatives, and link prediction focuses mainly on local structural patterns rather than global semantic alignment.

Instantiation of LDA. Guided by the above analysis, we instantiated a latent distribution alignment module to learn generalizable knowledge from multi-domain graphs. Specifically, we encode each graph $\mathcal{G}^i = (\hat{\mathbf{X}}^i,\mathbf{A}^i)$ using a single-layer GCN encoder with parameters shared across domains:

$$\mathbf{Z}^i = \text{GCN}_{\text{base}}(\hat{\mathbf{X}}^i,\mathbf{A}^i) = \text{ReLU}(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\hat{\mathbf{X}}^i\mathbf{W}), \tag{18}$$

where $\tilde{\mathbf{A}}^i = \mathbf{A}^i + \mathbf{I}^i$ denotes the adjacency matrix with self-loops, and $\tilde{\mathbf{D}}$ is the corresponding degree matrix. We treat $\mathbf{Z}^i$ as the semantic basis for approximating the domain-specific variational posterior $q_\phi(\mathbf{Z}^i \mid \hat{\mathbf{X}}^i)$. Following the variational autoencoder framework [16], we use two parallel GCNs (also shared across domains) to estimate the mean and log-variance:

$$\boldsymbol{\mu}^i = \mathrm{GCN}_\mu(\mathbf{Z}^i, \mathbf{A}^i), \quad \log \boldsymbol{\sigma}^i = \mathrm{GCN}_\sigma(\mathbf{Z}^i, \mathbf{A}^i), \tag{19}$$

where $\boldsymbol{\mu}^i$, $\boldsymbol{\sigma}^i$ represent the mean and variance of the domain-specific posterior distribution. These define a Gaussian posterior for each domain $\mathcal{G}^i$, formulated as $q_\phi(\mathbf{Z}^i \mid \hat{\mathbf{X}}^i) = \mathcal{N}(\mu^i, (\sigma^i)^2)$. To enable gradient backpropagation, we follow the previous work [16] and employ the reparameterization trick to sample latent codes from the variational distribution:

$$\mathbf{Z}^i = \boldsymbol{\mu}^i + \boldsymbol{\sigma}^i \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \tag{20}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ is standard Gaussian noise, and $\odot$ denotes element-wise multiplication. The latent codes are then decoded by a shared GCN decoder to reconstruct the DPU-aligned features:

$$\hat{\mathbf{X}}^i_{\mathrm{rec}} = \mathrm{Decoder}_\theta(\mathbf{Z}^i, \mathbf{A}^i), \tag{21}$$

where $\mathrm{Decoder}_\theta(\cdot)$ is a shared GCN-based decoder parameterized by θ. To learn a shared semantic distribution across domains, we optimize a multi-domain variational objective that jointly trains all graphs under a common generative prior. The objective is given by:

$$\mathcal{L}_{\mathrm{total}} = \sum_{i=1}^{M} \underbrace{\mathbb{E}_{q_\phi(\mathbf{Z}^i \mid \hat{\mathbf{X}}^i)} \left[\log p_\theta(\hat{\mathbf{X}}^i \mid \mathbf{Z}^i)\right]}_{\text{Domain-specific reconstruction}} - \underbrace{\mathrm{KL}\left(q_\phi(\mathbf{Z}^i \mid \hat{\mathbf{X}}^i) \,\|\, p(\mathbf{Z})\right)}_{\text{Cross-domain alignment}}, \tag{22}$$

where $p(\mathbf{Z})$ is a shared prior that acts as a universal reference for all domains. The gradient computed from $\mathcal{L}_{\mathrm{total}}$ is simultaneously propagated to both the GCN encoders and the transformation function in DPU. Although the mathematical form resembles the standard ELBO, its role in LEDA is fundamentally different. In VGAE [16], the KL term regularizes the latent space for reconstruction. In contrast, our KL term serves as a cross-domain alignment regularizer: by minimizing $\mathrm{KL}(q_\phi(\mathbf{Z}^i \mid \hat{\mathbf{X}}^i) \,\|\, p(\mathbf{Z}))$ for all domain i, we force each domain to adapt its posterior distribution to the same target. Combined with the shared encoder and DPU-aligned inputs, this mechanism enables the model to learn generalizable semantic representations across domains.

5 Experiments

In this section, we comprehensively evaluate the effectiveness and generalizability of the proposed LEDA by conducting comparisons with multiple baseline methods on both node-level and graph-level tasks. In Section 5.1, we introduce the experimental setup, while in Section 5.2, we analyze the model's performance across various tasks. Furthermore, in Section 5.3, we present ablation studies to examine the effectiveness of each component of LEDA.

5.1 Experimental Settings

Datsets. To effectively validate the performance of LEDA, we conduct experiments on eleven widely used benchmark datasets. We consider graph datasets spanning various domains, including: 1) *citation networks*: Cora, CiteSeer, and PubMed [30, 44]; 2) *co-purchase networks*: Photo and Computers [25]; 3)*co-author networks*: CS and Physics [31]; 4) *social network*: COLLAB, IMDB-BINARY [42]; 5) *biological network*: PROTEINS [5], ENZYMES [2].

Baselines. We compare LEDA's performance in four main categories as outlined below: 1) *Non-parametric models*: Raw features, SVD [32] and DeepWalk [27]; 2) *Graph self-supervised models*: GRACE [53], GCA [54], BGRL [35], DGI [37], GraphMAE [13]; 3) *Graph pre-train models*: OFA [22], GraphControl [52], GPPT [33], GraphPrompt [23], GPF [7], GCOPE [49], MDGPT [46], FUG [50], All-in-one [34], GPF-Plus [7], ULTRA(3g) [8], SCORE [39]; 4) *Graph semi-supervised models*: GCN [17].

Setups. To comprehensively evaluate the performance of our proposed method, we conduct extensive experiments under various settings as following: 1) *Cross-domain node classification*: In each experimental round, we designate one dataset as the target domain for downstream evaluation, while using the remaining six datasets as source domains for multi-domain pre-training. 2) *Cross-domain few-shot node classification*: To further evaluate the effectiveness of our approach in sample-scarce cross-domain settings, we conduct experiments under a few-shot node classification scenario. We consider 1-shot, 3-shot, and 5-shot settings, where each class in the target domain is provided with only 1, 3, or 5 labeled training examples, respectively.; 3) *Cross-domain zero-shot graph classification*: In addition to node-level tasks, we also perform cross-domain graph classification experiments on four datasets. Similar to the node classification task, we train the model on three of these datasets and test on the remaining one. More detailed implementation details can be found in Appendix A.1

5.2 Performance Analysis

Cross-domain node classification. To evaluate the effectiveness of LEDA in cross-domain node classification scenarios, we follow the setup of [50] and adopt a fixed set of hyper-parameters that are kept consistent across all training datasets. It is noteworthy that LEDA outperforms all in-domain baselines in cross-domain testing scenarios, indicating its strong ability to learn broad and transferable knowledge from multi-domain graphs and generalize to unseen domains. Moreover, while OFA and GraphControl are cross-domain methods capable of training on multi-domain graphs, their performance in cross-domain scenarios remains limited compared to LEDA, which can be attributed to the potential loss of critical information during feature alignment. OFA textualizes node attributes and leverages a large language model for encoding, whereas GraphControl employs kernel-based similarity to align node representations via a feature-driven adjacency matrix. Such data alignment strategies are sub-optimal when applied to highly complex features. Furthermore, recent advanced graph pre-training method FUG, despite designing a dimension encoder to learn knowledge lossless, still falls short in modeling shared semantics from multi-domain graphs. FUG employs a universal contrastive loss to constrain node distributions in representation space, which works well in in-domain settings but may become suboptimal for multi-domain knowledge learning. This further supports our analysis in Section 4.2. As shown in Table 1, LEDA outperforms FUG in cross-domain settings across six datasets, with particularly notable

Table 1: Accuracy (%) of cross-domain node classification with standard deviations. Each column represents a test domain, while others are train domains. CD means whether the model is trained on multi-domain datasets and tested on a cross-domain dataset. Aside from GCN, the best results are bolded and the second-best are underlined. Methods with * are reported from [50].

Method	CD	Cora	CiteSeer	PubMed	Photo	Computers	CS	Physics
Raw features*	×	57.90 ± 3.90	57.60 ± 2.85	79.55 ± 0.98	80.99 ± 1.65	75.59 ± 1.69	89.92 ± 0.95	93.18 ± 0.45
SVD*	×	56.36 ± 4.14	48.29 ± 3.18	82.80 ± 0.91	74.92 ± 1.82	71.26 ± 1.34	87.12 ± 0.73	92.48 ± 0.47
DeepWalk*	×	75.70 ± 0.00	50.50 ± 0.00	80.50 ± 0.00	89.44 ± 0.00	85.68 ± 0.00	84.61 ± 0.00	91.77 ± 0.00
GRACE*	×	83.20 ± 1.87	70.99 ± 2.29	85.46 ± 0.54	91.93 ± 0.83	85.36 ± 0.82	91.84 ± 0.37	OOM
GCA*	×	82.83 ± 2.29	72.06 ± 1.91	85.69 ± 0.68	92.63 ± 1.12	87.78 ± 0.78	92.69 ± 0.49	OOM
DGI*	×	83.24 ± 2.12	71.23 ± 2.37	84.62 ± 0.83	92.32 ± 0.49	86.12 ± 0.73	92.47 ± 0.60	94.47 ± 0.50
BGRL*	×	81.57 ± 2.07	70.10 ± 2.04	83.67 ± 0.84	92.34 ± 0.73	86.51 ± 1.53	92.12 ± 0.63	95.42 ± 0.41
GraphMAE	×	83.86 ± 0.62	72.26 ± 0.44	82.99 ± 0.21	92.21 ± 0.24	87.48 ± 0.28	92.28 ± 0.33	95.33 ± 0.21
FUG*	×	84.45 ± 2.45	72.43 ± 2.92	85.47 ± 1.13	93.07 ± 0.82	88.42 ± 0.98	92.89 ± 0.45	95.45 ± 0.27
OFA*	✓	75.90 ± 1.26	-	78.25 ± 0.71	-	-	-	-
GraphControl*	✓	-	-	-	89.66 ± 0.56	-	-	94.31 ± 0.12
FUG	✓	83.58 ± 1.21	69.65 ± 1.74	84.80 ± 0.30	92.51 ± 0.20	**89.29 ± 0.05**	92.45 ± 0.14	95.37 ± 0.33
LEDA (Ours)	✓	**84.71 ± 0.73**	**73.40 ± 0.15**	**87.10 ± 0.27**	**93.24 ± 0.98**	88.77 ± 0.56	**93.55 ± 0.42**	**95.70 ± 0.29**
Supervised GCN	×	82.80 ± 0.00	72.00 ± 0.00	84.80 ± 0.00	92.42 ± 0.00	86.51 ± 1.00	93.03 ± 0.00	95.65 ± 0.00

Table 2: Performance on cross-domain 1-shot node classification. Methods with * are reported from [46]. Each column represents a test domain, while others are train domains. The best results are bolded and the second-best are underlined.

Method \Test data	Cora	CiteSeer	PubMed	Photo	Computers	CS
DGI	24.19 ± 5.43	27.66 ± 4.93	45.77 ± 7.16	44.92 ± 7.71	27.92 ± 5.35	67.00 ± 6.72
BGRL	35.29 ± 6.54	34.99 ± 6.92	43.56 ± 7.54	48.25 ± 6.30	35.84 ± 8.74	62.55 ± 4.96
GraphMAE	35.99 ± 7.81	38.35 ± 8.36	49.23 ± 8.36	55.99 ± 9.62	45.17 ± 10.91	66.64 ± 7.03
GPPT*	15.37 ± 4.51	23.24 ± 2.94	36.56 ± 5.31	16.19 ± 4.73	19.22 ± 8.71	-
GraphPrompt*	35.90 ± 7.10	32.76 ± 7.66	43.34 ± 10.66	49.88 ± 8.31	43.03 ± 10.35	-
GPF*	37.84 ± 11.07	37.61 ± 8.87	46.36 ± 7.48	49.42 ± 7.04	37.00 ± 6.52	-
GCOPE*	33.38 ± 6.86	35.56 ± 6.81	42.10 ± 8.07	48.52 ± 7.78	40.22 ± 7.82	-
MDGPT*	42.26 ± 10.18	42.40 ± 9.26	49.82 ± 8.38	**64.82 ± 10.53**	49.77 ± 11.00	-
FUG	42.92 ± 11.51	36.78 ± 7.46	53.83 ± 8.77	62.80 ± 10.57	47.92 ± 11.51	68.32 ± 6.55
LEDA (Ours)	**50.70 ± 10.67**	**43.83 ± 10.04**	**54.34 ± 11.08**	64.35 ± 9.46	**51.00 ± 12.19**	**68.74 ± 8.00**

improvements on the CiteSeer dataset. This stems from the avoidance of modeling relative relationships among multi-domain graphs with entangled or inconsistent semantics. To further validate the transferability of LEDA, we also conducted experiments where pre-training was performed on the citation networks and testing was carried out on the co-purchase networks, and vice versa. The experimental results are presented in Appendix A.2.

Cross-domain few-shot node classification. In this scenario, we use fixed-parameters to pre-train the model and perform no fine-tuning or prompt-tuning. Specifically, we randomly select k nodes (k-shot) from each class and calculate the mean of their vectors as the class prototype vector and predict the node's class by calculating the similarity between the model output and the prototype vector. Unlike the above cross-domain node classification, the 1-shot setting places a stronger demand on the model's generalization ability. When k is set to 1,it can be observed from Table 2 that LEDA achieves the best results on five datasets and the second-best results on Photo. Notably, methods with $*$ incorporate prompt-tuning during downstream tasks, which are not currently used in LEDA. Moreover, FUG, which does not employ any fine-tuning or prompt-tuning strategies, performs well on most datasets; however, it shows suboptimal results on some datasets. This is due to its reliance on the quality of the sampled data. In contrast, LEDA does not require sampling from multi-domain data and achieves excellent performance across all datasets. Besides, we also conduct experiments with k set to 3 and 5, as illustrated in Figure 2. We observe that LEDA achieves the best performance in all datasets, demonstrating its exceptional generalization ability in few-shot scenarios. More experimental results in k-shot scenario can be found in Appendix A.2.

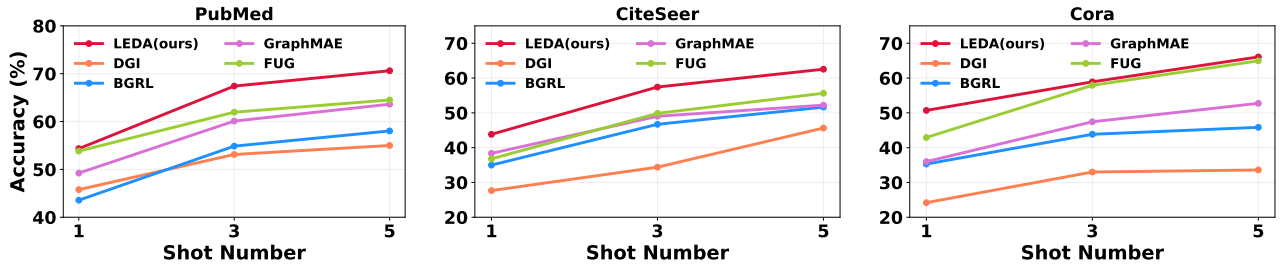

Figure 2: Performance on cross-domain few-shot node classification. The red line denotes our method. For traditional in-domain methods, we simply unify the input data dimensions by SVD.

Table 3: Performance on zero-shot graph classification. Methods with * are reported from [39]. The best results are bolded and the second-best are underline.

Method	IMDB-B		PROTEINS		ENZYMES		COLLAB	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
GPPT †*	50.15 ± 0.75	44.16 ± 6.70	60.92 ± 2.47	47.07 ± 11.95	21.29 ± 3.79	19.87 ± 2.99	47.18 ± 5.93	42.87 ± 7.70
All-in-one †*	60.07 ± 4.81	56.88 ± 0.80	66.49 ± 6.26	64.68 ± 5.35	23.96 ± 1.45	19.66 ± 3.11	51.66 ± 0.26	47.78 ± 0.10
GraphPrompt †*	54.75 ± 12.43	52.10 ± 13.61	59.17 ± 11.26	58.30 ± 10.88	22.29 ± 3.50	19.52 ± 3.36	48.25 ± 13.64	43.35 ± 10.75
GPF †*	59.65 ± 5.06	56.22 ± 6.17	63.91 ± 3.26	57.01 ± 5.79	22.00 ± 1.25	17.34 ± 2.45	47.42 ± 11.22	38.14 ± 0.44
GPF-Plus †*	57.93 ± 1.62	55.55 ± 2.03	62.92 ± 2.78	57.58 ± 7.28	22.92 ± 1.64	18.39 ± 2.76	47.24 ± 0.29	41.24 ± 0.31
ULTRA(3g) ‡*	49.25 ± 0.00	38.87 ± 0.00	58.09 ± 0.00	37.48 ± 0.00	15.21 ± 0.00	5.84 ± 0.00	64.53 ± 0.00	55.36 ± 0.00
SCORE ‡*	61.83 ± 1.60	60.91 ± 2.18	68.54 ± 1.47	65.23 ± 1.37	22.92 ± 2.03	21.77 ± 2.17	**65.45 ± 1.05**	**57.71 ± 1.82**
LEDA (Ours) ‡	**70.64 ± 1.91**	**70.60 ± 1.91**	**82.85 ± 1.02**	**82.67 ± 1.07**	**45.98 ± 2.03**	**40.07 ± 2.33**	55.83 ± 1.19	51.28 ± 1.59
GCN*	57.30 ± 0.98	54.62 ± 1.12	56.36 ± 7.97	46.69 ± 10.82	20.58 ± 2.00	15.25 ± 3.96	47.23 ± 0.61	41.10 ± 0.39

Cross-domain zero-shot graph classification. To enable a broader and fairer comparison of LEDA's performance in graph classification, we follow the evaluation setup from [39]. Methods with † refer to graph prompt-based algorithms under the 1-shot setting, while methods with ‡ represent algorithms under the 0-shot setting, without fine-tuning or prompt tuning. As shown in Table 3, LEDA significantly outperforms baselines on most datasets, demonstrating its ability to learn effective knowledge from multi-domain graphs. Notably, on the ENZYMES dataset, LEDA surpasses the second-best method by 23.06% in Accuracy, further highlighting its strong generalization capability. Moreover, it is worth noting that LEDA achieves moderate performance on COLLAB,in contrast to its consistently strong results in node classification. This may be due to COLLAB's comparatively denser topology relative to the other three datasets, as shown in Table 5.

5.3 Model Analysis

To evaluate the effectiveness of each component in LEDA, we conducted ablation studies on three datasets. We first removed the DPU and replaced it with a simple SVD-based data alignment approach. As shown in Table 4, LEDA without DPU results in a significant performance drop across all datasets. This supports our claim that while matrix factorization techniques can align the dimensions of different graph data, they may introduce redundant information or discard essential semantics, thereby hindering the learning of universal knowledge from multi-domain graphs. We then removed the LDA module, and similarly, LEDA without LDA also resulted in varying degrees of performance degradation across all datasets. This indicates that although DPU preserves as much essential information as possible from different graph datasets, its lack of cross-domain universal knowledge extraction leads to suboptimal performance. Furthermore, building on the removal of the LDA module, we introduced a contrastive loss to enhance the discriminability of representations in the unified space. Here, we use an InfoNCE-based contrastive loss, which aims to push anchor nodes away from negative samples. It is challenging to accurately define negative samples in the multi-domain pre-training setting; therefore, we adopt the mean feature of all nodes as the negative samples.

Table 4: Ablation studies on three cross-domain one-shot node classification datasets. CL means Contrastive Learning. The best result is bolded.

Method	Cora	CiteSeer	PubMed
w/o DPU	30.87 ± 7.47	36.55 ± 7.83	48.50 ± 9.23
w/o LDA	34.55 ± 7.24	41.07 ± 7.60	52.87 ± 8.26
DPU+CL	34.08 ± 7.12	40.85 ± 7.59	52.44 ± 8.30
LEDA	**50.70 ± 10.67**	**43.83 ± 10.04**	**54.34 ± 11.08**

By comparing DPU+CL with w/o LDA, we observe that the model performance remains nearly unchanged or slightly degrades after introducing contrastive loss. This aligns with our discussion in Section 4.2, suggesting that contrastive loss may lead to sub-optimal performance in multi-domain scenarios. Additional experimental results and detailed explanations are provided in Appendix A.2.

6 Conclusion

In this paper, we propose a novel universal graph pre-training model, LEDA, which effectively learns transferable knowledge from multi-domain graphs. By introducing a learnable domain projection unit, LEDA adaptively aligns features from multiple domains into a unified embedding space while preserving essential semantics and avoiding semantic conflicts. Furthermore, LEDA aligns the posterior distributions encoded from different domains with a shared prior distribution, enabling effective universal knowledge learning. Additionally, during distribution alignment, LEDA further guides the base vectors of domain-specific projections toward the shared semantic direction, thereby facilitating the learning of stable cross-domain representations. Extensive experimental results on multiple datasets and downstream tasks demonstrate the superior performance of LEDA. In few-shot cross-domain settings, LEDA outperforms existing methods by a considerable margin.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. 2005. Protein function prediction via graph kernels. *Bioinformatics* 21, suppl_1 (2005), i47–i56.

[3] Fenxiao Chen, Yun-Cheng Wang, Bin Wang, and C-C Jay Kuo. 2020. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing* 9 (2020), e15.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.

[5] Paul D Dobson and Andrew J Doig. 2003. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology* 330, 4 (2003), 771–783.

[6] Kaiwen Dong, Haitao Mao, Zhichun Guo, and Nitesh V Chawla. 2024. Universal Link Predictor By In-Context Learning on Graphs. *arXiv preprint arXiv:2402.07738* (2024).

[7] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. 2023. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems* 36 (2023), 52464–52489.

[8] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2023. Towards foundation models for knowledge graph reasoning. *arXiv preprint arXiv:2310.04562* (2023).

[9] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA*. 1024–1034.

[10] Dongxiao He, Yongqi Huang, Jitao Zhao, Xiaobao Wang, and Zhen Wang. 2025. Str-GCL: Structural Commonsense Driven Graph Contrastive Learning. In *Proceedings of the ACM on Web Conference 2025*. 1129–1141.

[11] Dongxiao He, Lianze Shan, Jitao Zhao, Hengrui Zhang, Zhen Wang, and Weixiong Zhang. 2024. Exploitation of a latent mechanism in graph contrastive learning: Representation scattering. *Advances in Neural Information Processing Systems* 37 (2024), 115351–115376.

[12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 639–648. doi:10.1145/3397271.3401063

[13] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. 2022. GraphMAE: Self-Supervised Masked Graph Autoencoders. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*. ACM, 594–604. https://doi.org/10.1145/3534678.3539321

[14] Yongqi Huang, Jitao Zhao, Dongxiao He, Di Jin, Yuxiao Huang, and Zhen Wang. 2025. Does GCL Need a Large Number of Negative Samples? Enhancing Graph Contrastive Learning with Effective and Efficient Negative Sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 17511–17518.

[15] Yongqi Huang, Jitao Zhao, Dongxiao He, Xiaobao Wang, Yawen Li, Yuxiao Huang, Di Jin, and Zhiyong Feng. 2025. One Prompt Fits All: Universal Graph Adaptation for Pretrained Models. *arXiv preprint arXiv:2509.22416* (2025).

[16] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *CoRR* abs/1611.07308 (2016). arXiv:1611.07308 http://arxiv.org/abs/1611.07308

[17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

[18] Gregory F Lawler and Vlada Limic. 2010. *Random walk: a modern introduction*. Vol. 123. Cambridge University Press.

[19] Jintang Li, Ruofan Wu, Wangbin Sun, Liang Chen, Sheng Tian, Liang Zhu, Changhua Meng, Zibin Zheng, and Weiqiang Wang. 2023. What's Behind the Mask: Understanding Masked Graph Modeling for Graph Autoencoders. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*. ACM, 1268–1279. https://doi.org/10.1145/3580305.3599546

[20] Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. 2024. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1725–1735.

[21] Mingkai Lin, Xiaobin Hong, Wenzhong Li, and Sanglu Lu. 2025. Unified Graph Neural Networks Pre-training for Multi-domain Graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 12165–12173.

[22] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. 2023. One for all: Towards training one graph model for all classification tasks. *arXiv preprint arXiv:2310.00149* (2023).

[23] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. 2023. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM web conference 2023*. 417–428.

[24] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[25] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[26] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[27] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*. ACM, 701–710. https://doi.org/10.1145/2623330.2623732

[28] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. 2019. On variational bounds of mutual information. In *International conference on machine learning*. PMLR, 5171–5180.

[29] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 1150–1160.

[30] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.

[31] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*. 243–246.

[32] Gilbert W Stewart. 1993. On the early history of the singular value decomposition. *SIAM review* 35, 4 (1993), 551–566.

[33] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. 2022. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1717–1727.

[34] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. 2024. All in One: Multi-task Prompting for Graph Neural Networks (Extended Abstract). In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*. ijcai.org, 8460–8465. https://www.ijcai.org/proceedings/2024/942

[35] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. 2021. Bootstrapped representation learning on graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*.

[36] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ

[37] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=rklz9iAcKQ

[38] Sebastian Vlaic, Theresia Conrad, Christian Tokarski-Schnelle, Mika Gustafsson, Uta Dahmen, Reinhard Guthke, and Stefan Schuster. 2018. ModuleDiscoverer: Identification of regulatory modules in protein-protein interaction networks. *Scientific reports* 8, 1 (2018), 433.

[39] Kai Wang and Siqiang Luo. 2024. Towards Graph Foundation Models: The Perspective of Zero-shot Reasoning on Knowledge Graphs. *arXiv preprint arXiv:2410.12609* (2024).

[40] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2021. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.* 32, 1 (2021), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

[41] Jun Xia, Yanqiao Zhu, Yuanqi Du, and Stan Z Li. 2022. A survey of pretraining on graphs: Taxonomy, methods, and applications. *arXiv preprint arXiv:2202.07893* (2022).

[42] Pinar Yanardag and SVN Vishwanathan. 2015. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1365–1374.

[43] Liangwei Yang, Zhiwei Liu, Yingtong Dou, Jing Ma, and Philip S Yu. 2021. Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation. In *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*. 2141–2145.

[44] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*. PMLR, 40–48.

[45] Xingtong Yu, Zechuan Gong, Chang Zhou, Yuan Fang, and Hui Zhang. 2025. SAMGPT: Text-free graph foundation model for multi-domain pre-training and cross-domain adaptation. In *Proceedings of the ACM on Web Conference 2025*. 1142–1153.

[46] Xingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. 2024. Text-free multi-domain graph pre-training: Toward graph foundation models. *arXiv preprint arXiv:2405.13934* (2024).

[47] Xiao-Meng Zhang, Li Liang, Lin Liu, and Ming-Jing Tang. 2021. Graph neural networks and their current applications in bioinformatics. *Frontiers in genetics* 12 (2021), 690049.

[48] Yuefeng Zhang. 2022. An Introduction to Matrix factorization and Factorization Machines in Recommendation System, and Beyond. *arXiv preprint arXiv:2203.11026* (2022).

[49] Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. 2024. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4443–4454.

[50] Jitao Zhao, Di Jin, Meng Ge, Lianze Shan, Xin Wang, Dongxiao He, and Zhiyong Feng. 2024. FUG: Feature-universal graph contrastive pre-training for graphs with diverse node features. *Advances in Neural Information Processing Systems* 37 (2024), 4003–4034.

[51] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI open* 1 (2020), 57–81.

[52] Yun Zhu, Yaoke Wang, Haizhou Shi, Zhenshuo Zhang, Dian Jiao, and Siliang Tang. 2024. Graphcontrol: Adding conditional control to universal graph pre-trained models for graph domain transfer learning. In *Proceedings of the ACM Web Conference 2024*. 539–550.

[53] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep Graph Contrastive Representation Learning. *CoRR* abs/2006.04131 (2020). arXiv:2006.04131 https://arxiv.org/abs/2006.04131

[54] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph Contrastive Learning with Adaptive Augmentation. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. ACM / IW3C2, 2069–2080. https://doi.org/10.1145/3442381.3449802

A Experimental Settings

A.1 Implementation Details

In this subsection, we elaborate on the details of our experimental implementation. In different scenarios, we follow different prior works, as each of them only covers a subset of our scenarios. For some methods whose reproduced performance was lower than reported, we instead report their best-known accuracy for fairness. All experiments were conducted on NVIDIA GeForce GTX 3090 GPU (24GB memory). **For the cross-domain node classification task**, we follow [35]. We first freeze the parameters of the pre-trained LEDA model and obtain node representations for the test data. Next, we train a downstream classifier using 10% of the test data, where the classifier is a simple linear model optimized with a logistic regression loss. We then evaluate the model on the remaining 90% of the data. We run LEDA 20 times and report the mean and standard deviation of the results. **For the cross-domain few-shot node classification task**, we follow [46]. We freeze the parameters of the pre-trained LEDA model and obtain node representations for the test data. For each class, we randomly select and label k samples, and compute their average as the class prototype vectors. Final predictions are made by measuring the similarity between each test node and the class prototypes. Moreover, it is worth noting that, compared to [46], our training domain includes the Co.CS dataset, which introduces greater domain diversity and poses a more challenging setting for evaluating the ability to learn generalizable knowledge across domains. To ensure statistical robustness, we run 500 times and report the mean and standard deviation. **For the cross-domain zero-shot graph classification task**, we follow [39]. We keep the pre-trained LEDA model frozen and derive graph-level representations via a pooling function. Predictions are made based on their similarity to class prototype vectors.

Moreover, for all the scenarios above, we adopt a single-layer GCN as the encoder for the LEDA model and use AdamW [24] as the optimizer. To ensure reproducibility, we fix the random seed to 66666. More detailed hyperparameter settings are provided in Table 6. Besides, in implementation, due to potentially differences in topologies across datasets, we apply different numbers of additional propagation steps after obtaining initial representations. In Table 6, Scenario 1 corresponds to cross-domain node classification. Scenarios 2 to 4 refer to cross-domain few-shot node classification with 1-shot, 3-shot, and 5-shot settings, respectively. Scenario 5 corresponds to cross-domain zero-shot graph classification. It is worth noting that we use a shared set of hyper-parameters across all datasets in each scenario.

A.2 Supplementary Experimental Results

Detailed values corresponding to the 3-shot and 5-shot plots in Figure 2. We present the detailed accuracy of LEDA under the 3-shot and 5-shot settings in Tables 7 and 8.

Supplementary cross-domain few-shot node classification results. Although the above few-shot cross-domain experiments demonstrate the strong performance of LEDA, the training and testing sets still involve datasets from similar domains, such as the citation networks (Cora, CiteSeer, PubMed) and the co-purchase networks (Photo, Computers). To better simulate real cross-domain scenarios, we construct a more challenging setting: using citation

Table 5: Dataset statistics of node and graph classification.

Node-level						
Dtasets	**Graphs**	**Nodes**	**Edges**	**Features**	**Classes**	**Domain**
Cora	1	2,708	4,732	1,433	7	Citation networks
CiteSeer	1	3,327	5,429	3,703	6	Citation networks
PubMed	1	19,717	44,338	500	3	Citation networks
Photo	1	7,650	119,081	745	8	Co-purchase networks
Computers	1	13,752	245,861	767	10	Co-purchase networks
CS	1	18,333	81,894	6,805	15	Co-author networks
Physics	1	34,493	247,962	8,415	5	Co-author networks
Graph-level						
Dtasets	**Graphs**	**Nodes**	**Edges**	**Features**	**Classes**	**Domain**
IMDB-BINARY	1,000	19.8	96.53	0	2	Social networks
COLLAB	5,000	74.5	2457.8	0	3	Social networks
PROTEINS	1,113	39.1	72.8	3	2	Biological networks
ENZYMES	600	32.6	62.1	3	6	Biological networks

Table 6: Hyper-parameters setting of LEDA.

Scenarios	Learning_rate	Epochs	DPU_Dim #1	DPU_Dim #2	DPU_Dim #3	LDA_Dim #1	LDA_Dim #2
Scenario 1	0.0001	2000	256	2048	1024	2048	1024
Scenario 2	0.0001	100	16	1024	512	1024	64
Scenario 3	0.0001	100	32	1024	1024	512	512
Scenario 4	0.0001	100	32	1024	1024	512	512
Scenario 5	0.0001	50	32	512	256	1024	1024

Table 7: Performance on cross-domain 3-shot node classification. Each column represents a test domain, while others are train domains. The best results are bolded and the second-best are underlined.

Method \Test data	Cora	CiteSeer	PubMed	Photo	Computers	CS
DGI	33.00 ± 4.79	34.39 ± 4.73	53.12 ± 6.73	56.88 ± 5.95	37.55 ± 5.49	76.44 ± 2.54
BGRL	43.84 ± 4.99	46.71 ± 5.41	54.86 ± 6.30	58.58 ± 6.43	45.65 ± 7.96	79.69 ± 2.73
GraphMAE	47.44 ± 6.18	49.02 ± 5.77	60.10 ± 6.43	67.96 ± 7.99	55.41 ± 10.07	81.95 ± 2.58
FUG	<u>57.92 ± 6.80</u>	<u>49.85 ± 5.45</u>	<u>61.96 ± 5.77</u>	**68.86 ± 7.39**	<u>56.20 ± 9.73</u>	<u>85.67 ± 1.85</u>
LEDA	**58.91 ± 7.15**	**57.42 ± 6.45**	**67.41 ± 7.91**	<u>68.39 ± 8.43</u>	**56.34 ± 12.05**	**86.75 ± 1.98**

networks as the training domain and evaluating on co-purchase and co-author networks, and vice versa. We conduct experiments under the 1-shot, 3-shot, and 5-shot settings, and report the corresponding results on Table 9.

B Supplementary of Discussion

Discussion of the potential semantic conflicts across domains. The potential semantic conflicts across domains has been acknowledged in previous work [46]. A key challenge in this context lies in the fact that features which are semantically meaningful in one domain may carry entirely different implications in another. For instance, a densely connected subgraph in a citation network like Cora may reflect topic coherence among papers, while in a co-purchase network such as Amazon Computers, a similar structure might arise from shared buying patterns that do not necessarily reflect category-level similarity. Such discrepancies can mislead the encoder during pre-training. This results in representations that conflate unrelated semantics, ultimately impairing their transferability and weakening downstream performance.

Discussion on the Reasonableness of the Distributional Assumption in Contrastive Learning. The assumption that the joint distribution $p(x, x')$ can be modeled via a similarity function

Table 8: Performance on cross-domain 5-shot node classification. Each column represents a test domain, while others are train domains. The best results are bolded and the second-best are underlined.

Method \Test data	Cora	CiteSeer	PubMed	Photo	Computers	CS
DGI	33.59 ± 3.97	45.64 ± 4.11	55.00 ± 5.15	59.04 ± 5.42	38.79 ± 5.42	79.63 ± 1.71
BGRL	45.84 ± 4.27	51.64 ± 3.85	58.04 ± 5.77	58.08 ± 5.11	48.71 ± 7.17	83.99 ± 2.24
GraphMAE	52.73 ± 4.66	52.22 ± 5.58	63.62 ± 5.28	68.48 ± 8.26	57.54±10.21	84.10 ± 1.71
FUG	64.95 ± 4.70	55.64 ±4.16	64.51 ± 4.84	70.27 ± 6.29	58.32 ± 8.64	88.04 ± 1.23
LEDA	**66.07 ± 5.72**	**62.54 ± 3.26**	**70.62 ± 6.07**	**71.08 ± 7.51**	**58.79 ± 11.8**	**88.47 ± 1.41**

Table 9: Performance on cross-domain few-shot node classification. The best results are bolded and the second-best are underlined.

Cross-domain 1-shot node classification						
Method \Test data	**Cora**	**CiteSeer**	**PubMed**	**Photo**	**Computers**	**CS**
DGI	27.93±4.83	26.33±4.08	40.62±5.39	34.50±6.82	24.12±5.00	33.88±6.14
BGRL	35.97±5.43	26.03±3.65	42.07±4.39	38.49±6.07	33.49±7.88	50.87±4.72
GraphMAE	37.67±7.79	36.43±8.45	48.89±9.20	55.80±9.48	43.22±11.55	66.45±7.58
FUG	40.88 ± 8.65	37.31 ± 7.50	50.38±9.60	60.46±10.38	45.72±11.72	72.57±8.60
LEDA	**50.51±10.94**	**44.83±10.05**	**54.18±11.03**	**63.95 ± 9.64**	**50.01±11.93**	**76.25±6.94**
Cross-domain 3-shot node classification						
DGI	35.36±4.67	30.90±3.64	42.79±4.72	48.56±6.02	27.80±4.70	66.67±3.02
BGRL	44.00±4.50	33.28±3.75	44.65±4.32	44.84±6.64	37.24±8.99	64.92±3.21
GraphMAE	50.33±5.96	45.58±6.68	58.77±6.55	62.33±7.66	55.76±8.35	84.43±1.83
FUG	51.62±7.15	48.40±5.65	60.52±6.52	65.75±7.33	56.61±9.80	85.97±1.95
LEDA	**60.05±7.90**	**57.76±6.04**	**64.47±7.76**	**66.77±8.48**	**58.36±10.96**	**86.43±2.12**
Cross-domain 5-shot node classification						
DGI	41.08±4.27	35.93±3.64	46.39±4.13	38.49±4.83	30.01±5.53	69.25±3.22
BGRL	50.08±3.86	37.18±3.43	48.78±4.05	46.68±6.10	41.63±8.62	77.95±2.26
GraphMAE	56.67±4.68	52.86±3.50	60.87±5.44	60.12±8.66	52.77±9.78	83.24±2.11
FUG	61.98±5.43	54.19±3.98	63.10±5.34	**70.34±6.20**	**62.11±8.49**	87.95±1.21
LEDA	**63.29±7.31**	**62.67±3.63**	**68.20±5.56**	66.43±7.90	60.60±10.62	**88.15±1.46**

Table 10: Supplementary ablation studies. The best results are bolded.

		3-shot			5-shot		
		Cora	**CiteSeer**	**PubMed**	**Cora**	**CiteSeer**	**PubMed**
Cross-domain	w/o DPU	44.07±6.39	50.91±5.29	60.79±7.29	50.56±5.27	56.56±3.72	65.88±5.48
	w/o LDA	45.84±6.01	52.46±5.24	62.25±6.41	51.07±4.48	57.32±3.67	65.70±4.85
	DPU+CL	44.98±6.01	52.15±5.27	61.52±6.43	50.01±4.58	57.02±3.71	65.08±4.92
	LEDA	**58.91±7.15**	**57.42±6.45**	**67.41±7.91**	**66.07±5.72**	**62.54±3.26**	**70.62±6.07**

$s_{ij} = f(x, x')$, and the marginal distribution $p(x)p(x')$ is approximated by $C \cdot e^{\xi(x,x')}$, serves as a theoretical foundation for analyzing contrastive learning objectives. This formulation aligns with the principle behind InfoNCE and related objectives, which aim to

maximize mutual information by contrasting positive and negative pairs. Specifically, interpreting the similarity function as a proxy for the log joint probability allows the contrastive loss to be viewed as an estimator of mutual information between different views of the same instance. The introduction of the bias term $\xi(x, x')$ offers additional flexibility, acknowledging the potential discrepancy between the assumed and true marginal distributions. While this term is not explicitly computed in practice, it enables a more precise theoretical analysis by isolating sources of deviation from ideal assumptions. Furthermore, similar assumptions have been employed in previous works to derive theoretical guarantees and bounds for representation quality and generalization. Although this modeling does not reflect an exact probabilistic characterization of the data, it provides a mathematically tractable abstraction that facilitates a deeper understanding of the learning dynamics. Therefore, this assumption is reasonable and widely accepted in the theoretical study of contrastive representation learning [28, 37, 53].

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009