Subgraph Gaussian Embedding Contrast for Self-Supervised Graph Representation Learning

Shifeng Xie, Aref Einizade, and Jhony H. Giraldo (✉)

LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France
{shifeng.xie, aref.einizade, jhony.giraldo}@telecom-paris.fr

Abstract. Graph Representation Learning (GRL) is a fundamental task in machine learning, aiming to encode high-dimensional graph-structured data into low-dimensional vectors. Self-Supervised Learning (SSL) methods are widely used in GRL because they can avoid expensive human annotation. In this work, we propose a novel Subgraph Gaussian Embedding Contrast (SubGEC) method. Our approach introduces a subgraph Gaussian embedding module, which adaptively maps subgraphs to a structured Gaussian space, ensuring the preservation of input subgraph characteristics while generating subgraphs with a controlled distribution. We then employ optimal transport distances, more precisely the Wasserstein and Gromov-Wasserstein distances, to effectively measure the similarity between subgraphs, enhancing the robustness of the contrastive learning process. Extensive experiments across multiple benchmarks demonstrate that SubGEC outperforms or presents competitive performance against state-of-the-art approaches. Our findings provide insights into the design of SSL methods for GRL, emphasizing the importance of the distribution of the generated contrastive pairs.

Keywords: Subgraph Gaussian embeddings · graph representation learning · self-supervised learning · optimal transport

1 Introduction

Graph Representation Learning (GRL) is a fundamental task in machine learning and data mining, aiming to encode high-dimensional, sparse graph-structured data into low-dimensional dense vectors [25]. Effective GRL techniques enable downstream applications such as node classification, link prediction, and community detection. Self-Supervised Learning (SSL) has emerged as a promising approach for GRL by reducing the dependence on extensive human annotation [22]. Among SSL methods, contrastive learning has gained significant attention due to its ability to learn meaningful representations by distinguishing similarities and differences among data samples. In contrastive learning, positive sample pairs typically consist of two augmented views of the same data point, which should be mapped close in the representation space, whereas negative sample pairs are formed by comparing different data points [5].

Existing graph-based contrastive learning methods primarily generate positive and negative pairs through structural perturbations [52,41,53] or learnable

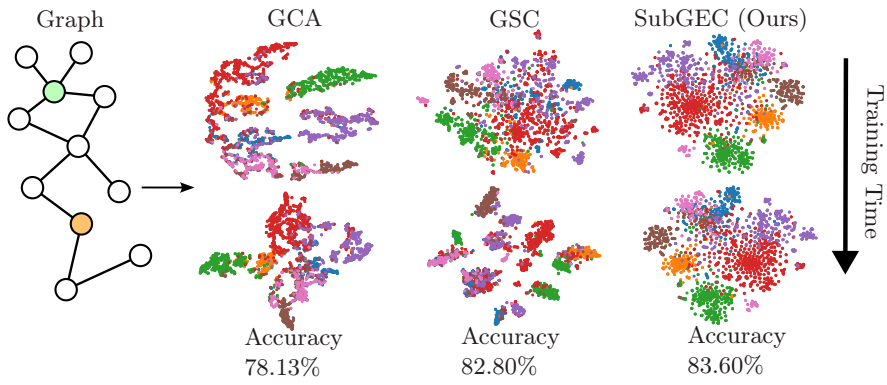


Fig. 1: t-stochastic neighbor embedding (t-SNE) visualizations of previous graph representation learning methods based on contrastive learning: GCA [53], GSC [17], and our method SubGEC. Each point corresponds to a node representation with reduced dimensionality, with colors indicating classes. Unlike GCA and GSC, which exhibit sharp boundaries, SubGEC maps node representations into a dense, uniform, and linearly separable space.

transformations [54,17]. However, Figure 1 shows t-stochastic neighbor embedding (t-SNE) visualizations of previous SSL methods, such as GCA [53] and GSC [17], where we observe uneven node distributions, sharp boundaries, and erroneous clusters. These issues suggest that existing approaches struggle to maintain smooth and meaningful representations, negatively impacting their performance in GRL tasks.

In this paper, we propose the **Sub**graph **G**aussian **E**mbedding **C**ontrast (SubGEC) model, a novel framework for graph contrastive learning. Our method introduces the Subgraph Gaussian Embedding (SGE) module, which maps input subgraphs to a structured Gaussian space, ensuring that the output features follow a Gaussian distribution using Kullback–Leibler (KL) divergence. This learnable mapping effectively controls the distribution of embeddings, improving representation quality. The generated subgraphs are then paired with the original subgraphs to form positive and negative contrastive pairs, and similarity is measured using Optimal Transport (OT) distances. By leveraging the Wasserstein and Gromov-Wasserstein distances, our approach enhances robustness and mitigates mode collapse (also called positive collapse [24]), where the embeddings shrink into a low-dimensional subspace, by controlling the embedding distribution.

Gaussian distributions provide several properties that make them useful in SSL for graphs. For example, they preserve important mathematical structures, such as displacement interpolation, which helps in clustering and interpolation tasks [51,13]. Gaussian smoothing also improves robustness by reducing noise and stabilizing learned representations [13]. Additionally, their simple parameterization using means and covariances makes them computationally efficient,

enabling scalability to high-dimensional spaces while avoiding excessive computational costs [51,6]. These properties make Gaussian-based OT a valuable tool in fields such as machine learning, physics, and statistical inference [10]. In this work, we theoretically and empirically prove the benefits of using Gaussian embeddings in contrastive learning.

The primary contributions of this paper are as follows:

- We introduce SubGEC, a novel framework that outperforms or remains competitive with state-of-the-art methods across eight benchmark datasets.
- We theoretically and empirically highlight the importance of mapping the distribution of contrastive pairs into a Gaussian space and analyze its impact on GRL.
- We conduct extensive ablation and validation studies to demonstrate the effectiveness of each component of SubGEC.

2 Related Work

GRL has gained significant attention due to its ability to encode structured data into meaningful representations. Here, we review the recent advancements in Graph Neural Networks (GNNs), SSL on graphs, and contrastive learning techniques.

GNNs [49] have been widely adopted for learning representations that capture both node features and graph topology [25]. Several architectures have been proposed to improve their learning capabilities. For example, Graph Convolutional Networks (GCNs) [29] leverage a simplification of graph filters to aggregate information from neighboring nodes. GraphSAGE [16] introduced an inductive learning framework with multiple aggregation functions, enabling generalization to unseen nodes. Furthermore, Graph Attention Networks (GAT) [47] integrate attention mechanisms to dynamically weigh node relationships, improving feature propagation. However, these models require supervised training.

SSL on graphs aims to design and solve learning tasks that do not require labeled data, avoiding costly supervised learning methodologies. Based on how these tasks are defined, SSL methods can be categorized into two main types: *predictive* and *contrastive* approaches.

Predictive methods focus on learning useful representations by generating perturbed versions of the input graph. For example, BGRL [41] learns node representations by encoding two perturbed versions of a graph using an online encoder and a target encoder. The online encoder is optimized to predict the target encoder's representation, while the target encoder is updated as an exponential moving average of the online encoder. BNLL [33] improves upon BGRL by introducing additional positive node pairs based on a homophily assumption, where neighboring nodes tend to share the same label. This is achieved by incorporating cosine similarity between a node's online representation and the weighted target representations of its neighbors. VGAE [28] adopts a variational autoencoder framework to reconstruct the input graph and its features.

Contrastive methods, which are the focus of this paper, generally outperform predictive methods in SSL for graphs. These methods can be classified based on how data pairs are defined: node-to-node, graph-to-graph, and node-to-graph comparisons. For example, GRACE [52] generates two perturbed graph views and applies contrastive learning at the node level. MUSE [50] refines this approach by extracting multiple embeddings—semantic, contextual, and fused—to enhance node-to-node contrastive learning. However, node-level contrastive learning is often suboptimal as it struggles to capture the overall structural information of the graph.

At the subgraph level, DGI [46] employs node-to-graph contrast, where it extracts node embeddings from the original and perturbed graphs and adjusts their agreement levels using a readout function. Spectral polynomial filter methods like GPR-GNN [7] and ChebNetII [19] offer greater flexibility than GCNs by adapting to different homophily levels. However, they often underperform when used as encoders for traditional SSL methods. To address this, PolyGCL [4] constrains polynomial filter expressiveness to construct high-pass and low-pass graph views while using a simple linear combination strategy for optimization. Unlike DGI, PolyGCL applies this contrastive approach to both high- and low-frequency embeddings extracted with shared-weight polynomial filters.

Subg-Con [23] extends DGI by performing contrastive learning at the subgraph level. It selects anchor nodes and extracts subgraphs using the personalized PageRank algorithm, adjusting the agreement between anchor nodes and their corresponding subgraphs for positive and negative pairs. However, methods like DGI and Subg-Con rely on a readout embedding to represent entire graphs, which disregards structural information. GSC [17] addresses this limitation by applying subgraph-level contrast using Wasserstein and Gromov-Wasserstein distances from OT to measure subgraph similarity, ensuring a more structurally-aware contrastive learning process. From another point of view, FOSSIL [38] fused Wasserstein and Gromov-Wasserstein distances [42,2] in the loss function to benefit from both node and subgraph-level features.

SubGEC leverages OT distance metrics to effectively measure subgraph dissimilarity as in [17]. However, unlike previous OT-based models such as GSC [17], our approach introduces a novel mapping of subgraphs into a structured Gaussian space. This design choice is driven by the properties of Gaussian embeddings, which enhance representation quality. Our work provides both theoretical justification and empirical validation for the effectiveness of this approach.

3 Preliminaries

3.1 Mathematical Notation

Consider an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. The feature matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times C}$ contains node features $\mathbf{x}_i \in \mathbb{R}^C$, where N is the number of nodes and C is the feature dimension. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ represents the graph topology, and $\mathbf{D}$ is the diagonal degree

matrix. For the i-th node, let $G^i = (\mathcal{V}^i, \mathcal{E}^i)$ be its induced Breadth-First Search [3] (BFS) subgraph with k^i nodes with adjacency matrix $\mathbf{A}^i \in \mathbb{R}^{k^i \times k^i}$ and feature matrix $\mathbf{X}^i \in \mathbb{R}^{k^i \times C}$. Our method embeds this subgraph (with the same sets of nodes and edges) producing adjacency matrix $\tilde{\mathbf{A}}^i$ and feature matrix $\tilde{\mathbf{X}}^i \in \mathbb{R}^{k^i \times F}$. In this work, we preserve the subgraph topology, so that $\tilde{\mathbf{A}}^i = \mathbf{A}^i$.

The KL divergence [44] is an asymmetry measure between two probability distributions P and Q. It quantifies the informational loss that occurs when distribution Q is utilized to approximate distribution P. The KL divergence is defined as:

$$D_{KL}(P\|Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right), \tag{1}$$

where $P(x)$ and $Q(x)$ are the probability masses of P and Q at each point x in the sample space $\mathcal{X}$.

3.2 Problem Formulation for Self-Supervised Graph Representation

The goal of self-supervised graph representation learning is to learn graph embeddings $\mathbf{R}$ through an encoder $\varepsilon : \mathbb{R}^{N \times N} \times \mathbb{R}^{N \times C} \to \mathbb{R}^{N \times F}$, where $\mathbf{R} = \varepsilon(\mathbf{A}, \mathbf{X}; \boldsymbol{\theta})$ is parametrized by some learnable parameters $\boldsymbol{\theta}$ and F represents the dimension of the embeddings (representation). This procedure is unsupervised, *i.e.*, it does not use labels. In this paper, $\varepsilon(\cdot)$ is a GNN [25], aiming to effectively capture both the graph's feature and topology information within the representation space.

3.3 Optimal Transport Distance

The Wasserstein distance [37], commonly used in OT, serves as a robust metric to compare the probability distributions defined over metric spaces. For subgraphs G^i and G^j, their corresponding feature matrices are denoted as $\mathbf{X}^i \in \mathbb{R}^{k^i \times C}$ and $\mathbf{X}^j \in \mathbb{R}^{k^j \times C}$. $\mathbf{x}^i_m \in \mathbb{R}^C$ and $\mathbf{x}^j_n \in \mathbb{R}^C$ respectively denote the feature vector of the m-th and n-th node in the subgraphs G^i and G^j, where $m = 1, 2, \ldots, k^i$ and $n = 1, 2, \ldots, k^j$. The r-Wasserstein distance between the feature distributions of these subgraphs is defined as [30,48]:

$$W_r(\mathbf{X}^i, \mathbf{X}^j) := \left(\min_{\mathbf{T} \in \pi(u,v)} \sum_{m=1}^{k^i} \sum_{n=1}^{k^j} \mathbf{T}_{(m,n)} d(\mathbf{x}^i_m, \mathbf{x}^j_n)^r \right)^{\frac{1}{r}}, \tag{2}$$

where $\pi(u, v)$ represents the set of all valid possible transport plans with probability distributions u and v responsible for generating $\mathbf{x}^i_m$ and $\mathbf{x}^j_n$, respectively. These distributions capture the node feature distributions in subgraphs G^i and G^j. The matrix $\mathbf{T} \in \pi(u, v)$ is the OT plan that matches the node pairs of the two subgraphs. $\mathbf{T}_{(m,n)}$ is value of the transportation plan between nodes m and n, and $d(\mathbf{x}^i_m, \mathbf{x}^j_n)$ represents a valid distance metric.

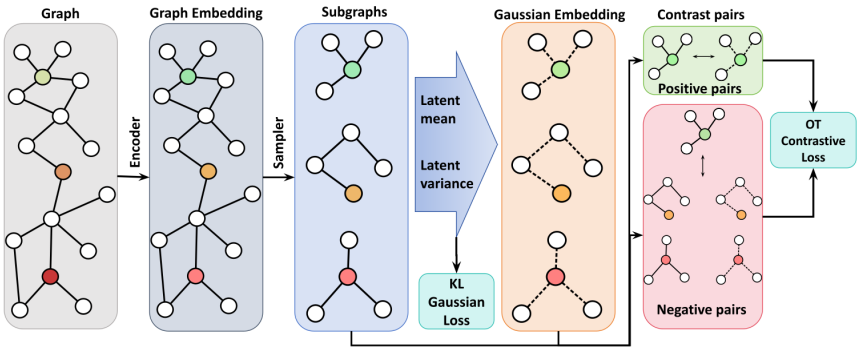


Fig. 2: Overview of the SubGEC method. Our model employs a graph encoder to obtain graph embeddings. We randomly select a set of nodes and then extract corresponding subgraphs using BFS sampling. Therefore, we use the proposed subgraph Gaussian embedding module using a KL loss to generate contrastive samples. Finally, we leverage OT distances for contrastive learning.

Similarly, the Gromov-Wasserstein distance [1,45] extends this idea to compare graph-structured data, where internal distances between nodes are taken into account. For two subgraphs G^i and G^j with adjacency matrices $\mathbf{A}^i$ and $\mathbf{A}^j$, and feature matrices $\mathbf{X}^i$ and $\mathbf{X}^j$, the Gromov-Wasserstein distance is defined as [45]:

$$\begin{aligned} &GW_r(\mathbf{A}^i, \mathbf{A}^j, \mathbf{X}^i, \mathbf{X}^j) \\ &:= \left(\min_{\mathbf{T} \in \pi(u,v)} \sum_{m,\tilde{m},n,\tilde{n}} \mathbf{T}_{(m,n)} \mathbf{T}_{(\tilde{m},\tilde{n})} \left| d(\mathbf{x}^i_m, \mathbf{x}^i_{\tilde{m}})^r - d(\mathbf{x}^j_n, \mathbf{x}^j_{\tilde{n}})^r \right| \right)^{\frac{1}{r}}, \end{aligned} \tag{3}$$

where $d(\mathbf{x}^i_m, \mathbf{x}^i_{\tilde{m}})$ and $d(\mathbf{x}^j_n, \mathbf{x}^j_{\tilde{n}})$ represent valid distance metrics between node pairs $(m, \tilde{m})$ in subgraph G^i, and $(n, \tilde{n})$ in subgraph G^j, respectively. Note that the node neighborhoods are considered by the term $\mathbf{T}$, thus relying on the graph topology.

In this work, for both the Wasserstein and Gromov-Wasserstein distances, we set $r = 1$ and define $d(\mathbf{x}^i_m, \mathbf{x}^j_n) = \exp\left(-\frac{\langle \mathbf{x}^i_m, \mathbf{x}^j_n \rangle}{\tau}\right)$, where $\langle \cdot, \cdot \rangle$ denotes the cosine similarity between node features, and τ is a temperature parameter.

4 Subgraph Gaussian Embedding Contrast (SubGEC)

Figure 2 shows an overview of our methodology, where our process begins with an encoder of the input graph. Subsequently, we obtain subgraphs utilizing BFS sampling. The embedded node representations within these subgraphs are thus embedded into a Gaussian latent space, enforced by the KL divergence regularization. Finally, we use the Wasserstein and Gromov-Wasserstein distances

to measure the dissimilarities in the subgraphs for contrastive learning. Our methodology is described in more detail in the following sections.

4.1 Graph Encoder

We begin by employing a graph encoder to preprocess the graph data [27,17]. The output feature matrix of the graph encoder is the desired graph representation. The graph encoder comprises some graph convolution layers. Further details on the implementation of these layers are provided in the Appendix A.

4.2 Subgraph Gaussian Embedding (SGE)

Constructing positive and negative pairs is crucial in graph contrastive learning [26]. The SGE module offers diversity to prevent mode collapse [24]. It also avoids generated subgraphs from becoming overly similar to the input subgraphs [14]. The SGE module comprises a GraphSAGE [31,16] network and then two GAT [47] models, representing the mean and variance for the KL loss. The first step in SGE is as follows:

$$\mathbf{H}_{\mathrm{GSA}} = \mathrm{GraphSAGE}\left(\mathbf{H}_{\mathrm{conv}}, \mathbf{A}\right), \tag{4}$$

where $\mathbf{H}_{\mathrm{conv}}$ represents the output of the graph encoder. Following GraphSAGE, GAT employs its attention mechanism to assign weights to the relationships between each node and its neighbors. The hidden means and variances are managed by separate GAT networks and processed as follows:

$$\boldsymbol{\mu} = \mathrm{GAT}_{\boldsymbol{\mu}}\left(\mathbf{H}_{\mathrm{GSA}}, \mathbf{A}\right), \quad \log \boldsymbol{\sigma} = \mathrm{GAT}_{\boldsymbol{\sigma}}\left(\mathbf{H}_{\mathrm{GSA}}, \mathbf{A}\right). \tag{5}$$

In this configuration, $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}$ are matrices of mean and variance vectors $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ for $i = 1, \ldots, N$, respectively. In our approach, the embedded matrix $\tilde{\mathbf{X}}$ is generated using the reparametrization trick [28] to facilitate the differentiation and optimization of our model as follows:

$$\tilde{\mathbf{X}} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \tag{6}$$

where $\odot$ states the element-wise multiplication, and the matrix $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_N]^\top$, where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for $i = 1, \ldots, N$, represents Gaussian (normal) noise.

4.3 Kullback-Leibler Gaussian Regularization Loss

In our approach, we introduce a regularization to the SGE module to guide the embedded subgraph node features toward a Gaussian distribution. This regularization is implemented using the KL divergence. The prior $p(\tilde{\mathbf{X}}) = \prod_{i=1}^{N} p(\tilde{\mathbf{x}}_i)$ is taken as a product of independent normal distributions for each latent variable $\tilde{\mathbf{x}}_i$, *i.e.*, the embedded feature vector of the i-th node. Similarly, by benefiting

from Gaussianity on the posterior distribution $q(\tilde{\mathbf{x}}_i|\mathbf{X},\mathbf{A}) = \mathcal{N}(\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2))$ [29], we express it on the whole data as:

$$q(\tilde{\mathbf{X}}|\mathbf{X},\mathbf{A}) = \prod_{i=1}^{N} q(\tilde{\mathbf{x}}_i|\mathbf{X},\mathbf{A}) = \prod_{i=1}^{N} \mathcal{N}(\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2)), \tag{7}$$

where $\mathrm{diag}(\mathbf{a})$ is a diagonal matrix with the elements of the vector $\mathbf{a}$ on its main diagonal, and $\boldsymbol{\sigma}_i^2$ obtains by element-wise power operation on the vector . The expression for the regularization then simplifies to (details in the Appendix B):

$$\mathcal{L}_R = \beta \ \mathrm{KL}\left(q(\tilde{\mathbf{X}}|\mathbf{X},\mathbf{A}) \| p(\tilde{\mathbf{X}})\right). \tag{8}$$

Here, β is a hyperparameter modulating the influence of the regularization term relative to the contrastive loss, which we introduce in Section 4.4, enabling precise control over the balance between data fidelity and distribution alignment.

4.4 Optimal Transport Contrastive and Model Loss

In terms of the architectures available for the contrastive learning loss function, options include the Siamese network loss [18], the triplet loss [20], and the noise contrastive estimation loss [15]. Given the presence of multiple sets of negative pairs in our model, we opt for the InfoNCE loss [34]. Our contrastive loss function integrates the Wasserstein and Gromov-Wasserstein distances into the InfoNCE loss formulation [34], addressing the complexities of graph-based data. The Wasserstein distance captures feature distribution representation within subgraphs. Furthermore, the Gromov-Wasserstein distance captures structural discrepancies, providing a topology-aware similarity measure. We define $W_{(\tau)}(\mathbf{X}^i, \tilde{\mathbf{X}}^i) := W(\mathbf{X}^i, \tilde{\mathbf{X}}^i)/\tau$, and $GW_{(\tau)}(\mathbf{A}^i, \mathbf{X}^i, \mathbf{A}^i, \tilde{\mathbf{X}}^i) := GW(\mathbf{A}^i, \mathbf{X}^i, \mathbf{A}^i, \tilde{\mathbf{X}}^i)/\tau$, where τ is a temperature hyperparameter. The Wasserstein ($\mathcal{L}_{\mathrm{W}}$) and Gromov-Wasserstein ($\mathcal{L}_{\mathrm{GW}}$) contrastive losses are given as follows:

$$\begin{aligned}
\mathcal{L}_{\mathrm{W}} &= -\sum_{i\in\mathcal{S}} \log \frac{e^{-W_{(\tau)}(\mathbf{X}^i, \tilde{\mathbf{X}}^i)}}{\sum_{j\in\mathcal{S}, j\neq i}^{N} \left(e^{-W_{(\tau)}(\mathbf{X}^i, \tilde{\mathbf{X}}^j)} + e^{-W_{(\tau)}(\mathbf{X}^i, \mathbf{X}^j)}\right)}, \\
\mathcal{L}_{\mathrm{GW}} &= -\sum_{i\in\mathcal{S}} \log \frac{e^{-GW_{(\tau)}(\mathbf{A}^i, \mathbf{X}^i, \mathbf{A}^i, \tilde{\mathbf{X}}^i)}}{\sum_{j\in\mathcal{S}, j\neq i}^{N} \left(e^{-GW_{(\tau)}(\mathbf{A}^i, \mathbf{X}^i, \mathbf{A}^j, \tilde{\mathbf{X}}^j)} + e^{-GW_{(\tau)}(\mathbf{A}^i, \mathbf{X}^i, \mathbf{A}^j, \mathbf{X}^j)}\right)},
\end{aligned} \tag{9}$$

where $\mathcal{S}$ is the set of sampled nodes. The model loss $\mathcal{L}$ incorporates the contrastive and regularization components as follows:

$$\mathcal{L} = \alpha\mathcal{L}_{\mathrm{W}} + (1-\alpha)\mathcal{L}_{\mathrm{GW}} + \mathcal{L}_{\mathrm{R}}, \tag{10}$$

where α is a hyperparameter that balances the emphasis on feature distribution and structural fidelity.

4.5 Theoretical Analysis of the Loss Function

The following theorem illustrates the effect of adding the term KL$(\cdot)$ to the overall loss function $\mathcal{L}$ in (10) with the input x and latent variable z.

Theorem 1. *By increasing the number of subgraphs (and consequently their associate node feature matrices), minimizing InfoNCE loss $\mathcal{L}_W(\cdot)$ in (9) and also the KL divergence in (10), the SubGEC model implicitly minimizes:*

$$\mathbb{E}_{\mathbf{X}\sim p(\mathbf{X}|\tilde{\mathbf{X}})}\left[KL\left(q_\phi(\tilde{\mathbf{X}}|\mathbf{X},\mathbf{A})||p(\tilde{\mathbf{X}}|\mathbf{X},\mathbf{A})\right)\right]. \tag{11}$$

Proof. Firstly, the following theorem from [34] outlines the relationship between minimizing the InfoNCE loss and maximizing mutual information between the input x and latent variable z, *i.e.*, $I(x,z)$.

Proposition 1 (From [34]). *The equivalence of maximizing the mutual information between the input x and latent variable z and minimizing $\mathcal{L}_{InfoNCE(N)}(x,z)$ becomes tighter by increasing the number of input data N as:*

$$I(x,z) \geq \log(N) - \mathcal{L}_{InfoNCE(N)}(x,z). \tag{12}$$

Next, by minimizing KL $(q_\phi(z|x)||p(z))$ leading to $q_\phi(z|x) \approx p(z)$, one can write:

$$\begin{aligned} I(x,z) &= \int\int p(x,z)\log\left(\frac{p(x,z)}{p(x)p(z)}\right)dx\,dz = \int\int \overbrace{p(x,z)}^{p(x|z)p(z)}\log\left(\frac{p(z|x)}{p(z)}\right)dx\,dz \\ &= \int p(x|z)\overbrace{\left[\int q_\phi(z|x)\log\left(\frac{p(z|x)}{q_\phi(z|x)}\right)dz\right]}^{-\mathrm{KL}(q_\phi(z|x)||p(z|x))}dx = -\mathbb{E}_{x\sim p(x|z)}\left[\mathrm{KL}\left(q_\phi(z|x)||p(z|x)\right)\right]. \end{aligned} \tag{13}$$

where we have used the mathematical expectation formula $\mathbb{E}_{x\sim p(x)}[f(x)] = \int f(x)p(x)dx$ for the last equality. Therefore, by increasing the number of inputs, minimizing $\mathcal{L}_{\mathrm{InfoNCE}(N)}(x,z)$ and also KL divergence KL $(q_\phi(z|x)||p(z))$, the network implicitly minimizes $\mathbb{E}_{x\sim p(x|z)}\left[\mathrm{KL}\left(q_\phi(z|x)||p(z|x)\right)\right]$, which means that the average distance over the samples from $p(x|z)$ between the parametric probability distribution $q_\phi(z|x)$ and $p(z|x)$ is minimized. Now, by replacing $\mathcal{L}_{\mathrm{InfoNCE}(N)}(x,z)$, $q_\phi(z|x)$, $p(z)$, $p(z|x)$, and $p(x|z)$ with $\mathcal{L}_W$, $q_\phi(\tilde{\mathbf{X}}|\mathbf{X},\mathbf{A})$, $p(\tilde{\mathbf{X}})$, $p(\tilde{\mathbf{X}}|\mathbf{X},\mathbf{A})$, and $p(\mathbf{X}|\tilde{\mathbf{X}})$, respectively, the proof is completed. □

SubGEC is driven by two key principles: (i) maximizing the mutual information between the input and latent variables and (ii) designing a robust encoder that generates latent embeddings closely aligned with the true latent distribution. Theorem 1 formally establishes that enforcing the joint minimization of the OT and KL losses in the overall loss (10) leads to the minimization of the expected KL divergence $\mathbb{E}_{\mathbf{X}\sim p(\mathbf{X}|\tilde{\mathbf{X}})}\left[\mathrm{KL}\left(q_\phi(\tilde{\mathbf{X}}|\mathbf{X},\mathbf{A})\|p(\tilde{\mathbf{X}}|\mathbf{X},\mathbf{A})\right)\right]$, ensuring an

Table 1: Overview of selected datasets used in the study.

Dataset	Nodes	Edges	Features	Avg. degree	Classes
Cora [29]	2,708	5,429	1,433	4.0	7
Citeseer [11]	3,312	4,732	3,703	2.9	6
Pubmed [39]	19,717	44,338	500	4.5	3
Coauthor [40]	18,333	163,788	6,805	17.9	15
Squirrel [36]	5,201	217,073	2,089	83.5	5
Chameleon [35]	2,277	36,101	2,325	31.7	5
Cornell [8]	183	298	1,703	3.3	5
Texas [8]	183	325	1,703	3.6	5

accurate estimation of the true conditional distribution $p(\tilde{\mathbf{X}}|\mathbf{X}, \mathbf{A})$. Simultaneously, this optimization strategy increases the mutual information between the input $\mathbf{X}$ and the latent embedding $\tilde{\mathbf{X}}$, thereby reinforcing the encoder's capacity to preserve essential input characteristics. Theorem 1 thus provides the theoretical foundation for SubGEC's design. Moreover, it highlights a crucial insight: minimizing the KL divergence alone does not necessarily maximize mutual information and may result in suboptimal performance, an observation we empirically validate in Section 5.2.

5 Experimental Evaluation

In this section, we present the empirical assessment of SubGEC by comparing its performance against current state-of-the-art methodologies across various public datasets. Additionally, through ablation studies, we verify the efficacy of our method. These studies analyze the contribution of individual SubGEC components to the overall performance. Finally, we analyze the computational cost to show our method's scalability to larger graphs. We also explore the sensitivity of the loss balance hyperparameter β and the size of the subgraph on the model's performance in Appendix C.

Datasets. We select several widely used datasets for graph node classification to evaluate SubGEC. These datasets encompass various types of networks, including academic citation networks, collaboration networks, and web page networks, providing diverse challenges and characteristics. Table 1 summarizes the basic statistics of these datasets.

Implementation details. We implement SubGEC using `PyG` and `PyTorch`. Our approach adopts a self-supervised scheme evaluated via linear probing. The model is trained using the official training subsets of the referenced datasets. Hyperparameter tuning involves a random search on the validation set to determine optimal values for the hyperparameters. The best configuration in validation is subsequently employed for tests on the dataset. We train our model with the Adam optimizer. We train our models on GPU architectures, including the RTX 3060 and A40.

Table 2: Performance comparison of self-supervised and supervised graph representation learning methods across eight benchmark datasets.

Method	Cora	Citeseer	Pubmed	Coauthor	Squirrel	Chameleon	Cornell	Texas
GCN	$81.40_{\pm 0.50}$	$70.30_{\pm 0.50}$	$76.80_{\pm 0.70}$	$93.03_{\pm 0.31}$	$53.43_{\pm 1.52}$	$64.82_{\pm 2.32}$	$60.54_{\pm 3.30}$	$67.57_{\pm 4.80}$
GAT	$83.00_{\pm 0.52}$	$72.50_{\pm 0.30}$	$79.00_{\pm 0.24}$	$92.31_{\pm 0.24}$	$42.72_{\pm 3.27}$	$63.90_{\pm 2.19}$	$76.00_{\pm 3.63}$	$78.87_{\pm 3.78}$
MUSE	$69.90_{\pm 0.41}$	$66.35_{\pm 0.40}$	$79.95_{\pm 0.59}$	$90.75_{\pm 0.39}$	$40.15_{\pm 3.04}$	$55.59_{\pm 2.21}$	$83.78_{\pm 3.42}$	$83.78_{\pm 2.79}$
POLYGCL	$\mathbf{84.89}_{\pm 0.62}$	$\mathbf{76.28}_{\pm 0.85}$	$81.02_{\pm 0.27}$	$93.76_{\pm 0.08}$	$55.29_{\pm 0.72}$	$\mathbf{71.62}_{\pm 0.96}$	$77.86_{\pm 3.11}$	$85.24_{\pm 1.80}$
GREET	$84.40_{\pm 0.77}$	$74.10_{\pm 0.44}$	$80.29_{\pm 0.24}$	$\mathbf{94.65}_{\pm 0.18}$	$39.76_{\pm 0.75}$	$60.57_{\pm 1.03}$	$78.36_{\pm 3.77}$	$78.03_{\pm 3.94}$
GRACE	$83.30_{\pm 0.74}$	$72.10_{\pm 0.60}$	$\mathbf{86.70}_{\pm 0.16}$	$92.78_{\pm 0.04}$	$52.10_{\pm 0.94}$	$52.29_{\pm 1.49}$	$60.66_{\pm 11.32}$	$75.74_{\pm 2.95}$
GSC	$82.80_{\pm 0.10}$	$71.00_{\pm 0.10}$	$85.60_{\pm 0.20}$	$91.88_{\pm 0.11}$	$51.32_{\pm 0.21}$	$64.02_{\pm 0.29}$	$93.56_{\pm 1.73}$	$88.64_{\pm 1.21}$
DGI	$81.99_{\pm 0.95}$	$71.76_{\pm 0.80}$	$77.16_{\pm 0.24}$	$92.15_{\pm 0.63}$	$38.80_{\pm 0.76}$	$58.00_{\pm 0.70}$	$70.82_{\pm 7.21}$	$81.48_{\pm 2.79}$
GCA	$78.13_{\pm 0.85}$	$67.81_{\pm 0.75}$	$80.63_{\pm 0.31}$	$93.10_{\pm 0.20}$	$47.13_{\pm 0.61}$	$56.54_{\pm 1.07}$	$53.11_{\pm 9.34}$	$81.02_{\pm 2.30}$
GraphMAE	$84.20_{\pm 0.40}$	$73.40_{\pm 0.40}$	$81.10_{\pm 0.40}$	$80.63_{\pm 0.15}$	$48.26_{\pm 1.21}$	$71.05_{\pm 0.36}$	$61.93_{\pm 4.59}$	$67.80_{\pm 3.37}$
SubGEC	$83.60_{\pm 0.10}$	$73.14_{\pm 0.14}$	$84.60_{\pm 0.10}$	$92.34_{\pm 0.04}$	$\mathbf{56.39}_{\pm 0.57}$	$69.14_{\pm 1.12}$	$\mathbf{94.57}_{\pm 2.13}$	$\mathbf{92.38}_{\pm 0.81}$

Hyperparameter random search. Informed by the findings of [9][43][12], which indicate the sensitivity of GNNs to hyperparameter settings, we undertake random searches for hyperparameter optimization. The training proceeds on the official splits of the train datasets, with the random search conducted on the validation dataset to pinpoint the best configurations. These settings are then implemented to evaluate the model on the test dataset. The ranges of the hyperparameters explored and our code implementation are available[1] and will be made public after acceptance to facilitate replication and further research.

5.1 Classification Results

In our study, we compared our model against five state-of-the-art self-supervised node classification algorithms: POLYGCL [4], GREET [32], GRACE [52], GSC [17], and MUSE [50]. Additionally, we include three classic SSL algorithms for a comprehensive comparison: DGI [46], GCA [53], and GraphMAE [21]. To provide a broader context, we also report the training results from two supervised learning models: GCN [29] and GAT [47].

The results in Table 2 highlight the strong performance of SubGEC across diverse datasets. Our model outperforms other state-of-the-art algorithms on three out of eight benchmarks while demonstrating competitive results on the remaining datasets. Notably, it achieves the highest accuracy on the strongly heterophilic Squirrel, Cornell, and Texas datasets, exceeding GSC, POLYGCL, and other baselines. This suggests that the proposed design is particularly robust in settings where node connectivity patterns deviate from typical homophilic assumptions. Although POLYGCL slightly surpasses SubGEC on certain homophilic datasets (*e.g.*, Cora and Citeseer), SubGEC remains comparably strong there. Overall, these results highlight SubGEC's robustness in handling heterophilic structures while maintaining strong performance on homophilic graphs, demonstrating its versatility across diverse graph topologies.

[1] `https://github.com/ShifengXIE/SubGEC/tree/main`

Table 3: Ablation study on KL regularization and other components. **Reg.** denotes the type of regularization applied, with possible choices including no regularization (✗), KL divergence (**KL**), and dropout (**D.**). **L1** indicates whether the L1 norm was used as a reconstruction loss. **De.** represents whether a decoder was included in the model. **Cons.** refers to whether contrastive loss was incorporated.

Reg.	L1	De.	Cons.	Cora	Citeseer	Pubmed	Coauthor	Squirrel	Chameleon	Cornell	Texas
✗	✗	✗	✔	83.00$_{\pm 0.07}$	71.88$_{\pm 0.07}$	**85.46**$_{\pm 0.04}$	91.96$_{\pm 0.09}$	42.99$_{\pm 0.17}$	64.25$_{\pm 0.21}$	**94.58**$_{\pm 0.22}$	75.72$_{\pm 0.40}$
KL	✔	✗	✗	78.78$_{\pm 1.09}$	68.33$_{\pm 1.00}$	75.56$_{\pm 1.65}$	88.86$_{\pm 0.25}$	30.52$_{\pm 0.48}$	48.12$_{\pm 0.63}$	68.43$_{\pm 0.55}$	73.83$_{\pm 1.20}$
KL	✗	✔	✔	82.80$_{\pm 0.07}$	73.00$_{\pm 0.08}$	80.26$_{\pm 0.37}$	92.03$_{\pm 0.13}$	35.47$_{\pm 0.21}$	60.30$_{\pm 0.21}$	93.56$_{\pm 0.64}$	88.98$_{\pm 0.52}$
KL	✔	✗	✔	81.60$_{\pm 0.99}$	69.60$_{\pm 0.10}$	67.54$_{\pm 0.35}$	87.03$_{\pm 0.65}$	30.52$_{\pm 0.48}$	43.26$_{\pm 0.66}$	53.29$_{\pm 0.13}$	63.45$_{\pm 0.48}$
D.	✗	✗	✔	79.00$_{\pm 0.21}$	70.60$_{\pm 3.52}$	80.84$_{\pm 0.05}$	91.52$_{\pm 0.37}$	44.24$_{\pm 0.50}$	58.94$_{\pm 0.72}$	85.76$_{\pm 0.24}$	87.98$_{\pm 0.15}$
KL	✗	✗	✔	**83.60**$_{\pm 0.10}$	**73.14**$_{\pm 0.14}$	84.60$_{\pm 0.10}$	**92.34**$_{\pm 0.04}$	**56.39**$_{\pm 0.57}$	**69.14**$_{\pm 1.12}$	94.57$_{\pm 2.13}$	**92.38**$_{\pm 0.81}$

5.2 Ablation Studies

KL divergence and contrastive loss. The first ablation study is concerned with analyzing some elements of SubGEC such as the architectural choices, the KL regularization, and the contrastive loss. The outcomes of this ablation study are presented in Table 3.

The first row in Table 3 analyzes the case where we drop the KL loss from SubGEC. We observe that in overall the performance decreases, demonstrating the importance of the KL loss as theoretically proved in Section 4.5. On the contrary, the second row in Table 3 includes only the KL loss and L1 reconstruction loss without including the contrastive loss. This effectively models a Variational Autoencoder (VAE) type method, where we observe a loss in performance. This result also aligns with the theoretical findings in Theorem 1, where we show that solely relying on the minimization of the KL loss does not guarantee the accurate estimation of the encoder distribution and can lead to performance degradation compared to using both the KL and contrastive loss functions.

The third model in Table 3 incorporates a decoder into SubGEC, *i.e.*, we use a VAE-type architecture to generate contrastive pairs. The decoder consists of two fully connected multi-layer perceptrons. This model achieves competitive results only on specific databases, illustrating that SubGEC is not merely a combination of a VAE generative model and contrastive learning training methodologies. The fourth model includes a norm-1 reconstruction loss, calculated as the norm of the difference between input and output features, adding a constraint to enforce similarity between input and output features. The results indicate that enforcing such similarity is not reasonable. Finally, the fifth model in Table 3 replaces the KL divergence with the commonly used regularization technique, dropout. The results show that our method outperforms dropout.

Contrastive loss. The second ablation study examines the impact of the distance metric used in our contrastive loss, specifically comparing OT distances with alternative approaches. Table 4 presents the results of this study, evaluating models with Wasserstein-only, Gromov-Wasserstein-only, and L1-only metrics,

Table 4: Ablation studies on the choice of the distance metric in the contrastive loss. **W** indicates the use of the Wasserstein distance. **GW** indicates the use of the Gromov-Wasserstein distance. **L1** indicates the use of a simple L1 distance.

W	GW	L1	Cora	Citeseer	Pubmed	Coauthor	Squirrel	Chameleon	Cornell	Texas
✔	✗	✗	$77.00_{\pm 0.81}$	$66.80_{\pm 1.39}$	$78.24_{\pm 1.22}$	$88.65_{\pm 0.62}$	$49.02_{\pm 0.85}$	$62.50_{\pm 0.49}$	$91.29_{\pm 0.10}$	$87.67_{\pm 0.09}$
✗	✔	✗	$76.20_{\pm 1.56}$	$68.98_{\pm 0.23}$	$80.20_{\pm 1.42}$	$91.08_{\pm 0.28}$	$45.16_{\pm 0.55}$	$56.17_{\pm 0.28}$	$90.16_{\pm 1.52}$	$88.47_{\pm 0.89}$
✗	✗	✔	$79.84_{\pm 0.68}$	$69.80_{\pm 0.64}$	$79.20_{\pm 2.54}$	$82.23_{\pm 1.51}$	$47.10_{\pm 0.58}$	$58.35_{\pm 0.92}$	$90.33_{\pm 1.08}$	$84.59_{\pm 0.80}$
✔	✔	✗	$\mathbf{83.60}_{\pm 0.10}$	$\mathbf{73.14}_{\pm 0.14}$	$\mathbf{84.60}_{\pm 0.10}$	$\mathbf{92.34}_{\pm 0.04}$	$\mathbf{56.39}_{\pm 0.57}$	$\mathbf{69.14}_{\pm 1.12}$	$\mathbf{94.57}_{\pm 2.13}$	$\mathbf{92.38}_{\pm 0.81}$

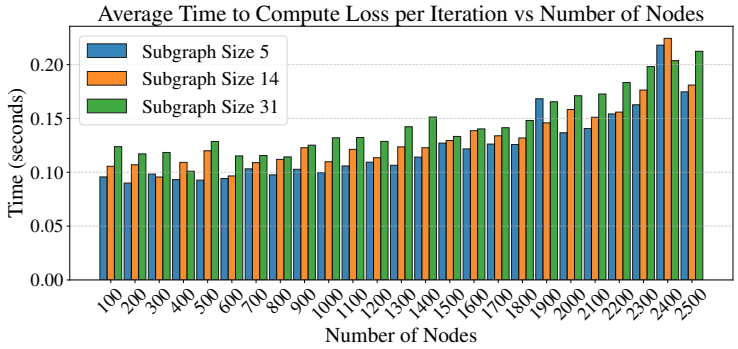


Fig. 3: Average time to compute loss per iteration as a function of the number of nodes. The figure compares the computation times for three different subgraph sizes (5, 14, and 31).

as well as SubGEC. Our findings indicate that excluding OT distances leads to suboptimal performance, particularly on heterophilic datasets. Additionally, we observe that the Gromov-Wasserstein distance slightly outperforms the Wasserstein distance on homophilic datasets. Most importantly, incorporating both Wasserstein and Gromov-Wasserstein distances in the contrastive loss consistently yields the best performance across all datasets.

5.3 Running Time

We employ a subgraph sampling strategy to avoid the high computational complexity of OT computations. Figure 3 shows the average time to compute the loss per iteration. The running time can vary due to server performance fluctuations, leading to non-monotonic timing variations. We observe that the running time remains low, increasing modestly as the graph size grows from 100 to 2,500 nodes, with times ranging from 0.1 to 0.2 seconds. We attribute this increase in computational time to the higher dimensionalities of the adjacency matrices when subgraphs are sampled. Overall, SubGEC keeps a low running time even for increasing graph sizes, potentially enabling applications in large-scale graph SSL tasks.

6 Conclusion

This paper introduces the SubGEC, a novel GRL framework that leverages subgraph Gaussian embeddings for self-supervised contrastive learning. Our approach maps subgraphs into a Gaussian space, ensuring a controlled distribution while preserving essential subgraph characteristics. We also incorporate the OT Wasserstein and Gromov-Wasserstein distances into our contrastive loss. From a theoretical perspective, we demonstrated that our method minimizes the KL divergence between the learned encoder distribution and the Gaussian distribution while maximizing mutual information between input and latent variables. Our experiments on multiple benchmark datasets validate these theoretical insights and show that SubGEC outperforms or presents competitive performance against previous state-of-the-art models. Our findings emphasize the importance of controlling the distribution of contrastive pairs in SSL.

Acknowledgment

This research was supported by DATAIA Convergence Institute as part of the «Programme d'Investissement d'Avenir», (ANR-17-CONV-0003) operated by the center Hi! PARIS. This work was also supported by the ANR French National Research Agency under the JCJC projects DeSNAP (ANR-24CE23-1895-01).

References

1. Arya, S., Auddy, A., Clark, R.A., Lim, S., Memoli, F., Packer, D.: The Gromov–Wasserstein distance between spheres. Foundations of Computational Mathematics pp. 1–56 (2024)
2. Brogat-Motte, L., Flamary, R., Brouard, C., Rousu, J., d'Alché Buc, F.: Learning to predict graphs with fused Gromov-Wasserstein barycenters. In: International Conference on Machine Learning (2022)
3. Bundy, A., Wallen, L.: Breadth-first search. Catalogue of Artificial Intelligence Tools pp. 13–13 (1984)
4. Chen, J., Lei, R., Wei, Z.: PolyGCL: Graph contrastive learning via learnable spectral polynomial filters. In: International Conference on Learning Representations (2024)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning (2020)
6. Chen, Y., Georgiou, T.T., Tannenbaum, A.: Optimal transport for Gaussian mixture models. IEEE Access **7**, 6269–6278 (2018)
7. Chien, E., Peng, J., Li, P., Milenkovic, O.: Adaptive universal generalized PageRank graph neural network. In: International Conference on Learning Representations (2021)
8. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S.: Learning to extract symbolic knowledge from the World Wide Web. AAAI Conference on Artificial Intelligence (1998)

9. Gasteiger, J., Weiß enberger, S., Günnemann, S.: Diffusion improves graph learning. In: Advances in Neural Information Processing Systems (2019)
10. Genevay, A., Cuturi, M., Peyré, G., Bach, F.: Stochastic optimization for large-scale optimal transport. In: Advances in Neural Information Processing Systems (2016)
11. Giles, C.L., Bollacker, K.D., Lawrence, S.: CiteSeer: An automatic citation indexing system. In: ACM Conference on Digital Libraries (1998)
12. Giraldo, J.H., Skianis, K., Bouwmans, T., Malliaros, F.D.: On the trade-off between over-smoothing and over-squashing in deep graph neural networks. In: ACM International Conference on Information and Knowledge Management (2023)
13. Goldfeld, Z., Greenewald, K.: Gaussian-smoothed optimal transport: Metric structure and statistical efficiency. In: International Conference on Artificial Intelligence and Statistics (2020)
14. Grill, J.B., et al.: Bootstrap your own latent: A new approach to self-supervised learning. In: Advances in Neural Information Processing Systems (2020)
15. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: International Conference on Artificial Intelligence and Statistics (2010)
16. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems (2017)
17. Han, Y., Hui, L., Jiang, H., Qian, J., Xie, J.: Generative subgraph contrast for self-supervised graph representation learning. In: European Conference on Computer Vision (2022)
18. He, A., Luo, C., Tian, X., Zeng, W.: A twofold siamese network for real-time object tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018)
19. He, M., Wei, Z., Wen, J.R.: Convolutional neural networks on graphs with Chebyshev approximation, revisited. In: Advances in Neural Information Processing Systems (2022)
20. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 (2017)
21. Hou, Z., Liu, X., Cen, Y., Dong, Y., Yang, H., Wang, C., Tang, J.: GraphMAE: Self-supervised masked graph autoencoders. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2022)
22. Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F.: A survey on contrastive self-supervised learning. Technologies **9**(1), 2 (2020)
23. Jiao, Y., Xiong, Y., Zhang, J., Zhang, Y., Zhang, T., Zhu, Y.: Sub-graph contrast for scalable self-supervised graph representation learning. In: IEEE International Conference on Data Mining (2020)
24. Jing, L., Vincent, P., LeCun, Y., Tian, Y.: Understanding dimensional collapse in contrastive self-supervised learning. In: International Conference on Learning Representations (2022)
25. Ju, W., Fang, Z., Gu, Y., Liu, Z., Long, Q., Qiao, Z., Qin, Y., Shen, J., Sun, F., Xiao, Z., et al.: A comprehensive survey on deep graph representation learning. Neural Networks (2024)
26. Ju, W., Wang, Y., Qin, Y., Mao, Z., Xiao, Z., Luo, J., Yang, J., Gu, Y., Wang, D., Long, Q., Yi, S., Luo, X., Zhang, M.: Towards graph contrastive learning: A survey and beyond. arXiv preprint arXiv:2405.11868 (2024)
27. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations (2014)

28. Kipf, T.N., Welling, M.: Variational graph auto-encoders. In: Advances in Neural Information Processing Systems - Workshop (2016)
29. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017)
30. Kolouri, S., Park, S.R., Thorpe, M., Slepcev, D., Rohde, G.K.: Optimal mass transport: Signal processing and machine-learning applications. IEEE Signal Processing Magazine **34**(4), 43–59 (2017)
31. Liu, J., Ong, G.P., Chen, X.: GraphSAGE-based traffic speed forecasting for segment network with sparse data. IEEE Transactions on Intelligent Transportation Systems **23**(3), 1755–1766 (2020)
32. Liu, Y., Zheng, Y., Zhang, D., Lee, V.C., Pan, S.: Beyond smoothing: Unsupervised graph representation learning with edge heterophily discriminating. In: Proceedings of the AAAI Conference on Artificial Intelligence (2023)
33. Liu, Y., Zhang, H., He, T., Zheng, T., Zhao, J.: Bootstrap latents of nodes and neighbors for graph self-supervised learning. In: European Conference on Machine Learning and Knowledge Discovery in Databases (2024)
34. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
35. Pei, H., Wei, B., Chang, K.C.C., Lei, Y., Yang, B.: Geom-GCN: Geometric graph convolutional networks. In: International Conference on Learning Representations (2020)
36. Rozemberczki, B., Allen, C., Sarkar, R.: Multi-scale attributed node embedding. Journal of Complex Networks **9**(2) (2021)
37. Rüschendorf, L.: The Wasserstein distance and approximation theorems. Probability Theory and Related Fields (1985)
38. SANGARE, A.S., Dunou, N., Giraldo, J.H., Malliaros, F.D.: A fused Gromov-Wasserstein approach to subgraph contrastive learning. Transactions on Machine Learning Research (2025)
39. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI Magazine **29**(3), 93–93 (2008)
40. Shchur, O., Mumme, M., Bojchevski, A., Günnemann, S.: Pitfalls of graph neural network evaluation. In: Advances in Neural Information Processing Systems - Workshops (2018)
41. Thakoor, S., Tallec, C., Azar, M.G., Azabou, M., Dyer, E.L., Munos, R., Veličković, P., Valko, M.: Large-scale representation learning on graphs via bootstrapping. In: International Conference on Learning Representations (2021)
42. Titouan, V., Courty, N., Tavenard, R., Flamary, R.: Optimal transport for structured data with application on graphs. In: International Conference on Machine Learning. pp. 6275–6284 (2019)
43. Topping, J., Giovanni, F.D., Chamberlain, B.P., Dong, X., Bronstein, M.M.: Understanding over-squashing and bottlenecks on graphs via curvature. In: International Conference on Learning Representations (2022)
44. Van Erven, T., Harremos, P.: Rényi divergence and Kullback-Leibler divergence. IEEE Transactions on Information Theory **60**(7), 3797–3820 (2014)
45. Vayer, T., Chapel, L., Flamary, R., Tavenard, R., Courty, N.: Fused Gromov-Wasserstein distance for structured objects. Algorithms **13**(9), 212 (2020)
46. Veličković, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. In: International Conference on Learning Representations (2019)
47. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)

48. Villani, C.: Topics in optimal transportation, vol. 58. American Mathematical Soc. (2021)
49. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems **32**(1), 4–24 (2020)
50. Yuan, M., Chen, M., Li, X.: MUSE: Multi-view contrastive learning for heterophilic graphs. In: ACM International Conference on Information and Knowledge Management (2023)
51. Zhu, J., Xu, K., Tannenbaum, A.: Optimal transport for vector Gaussian mixture models. In: Advances in Neural Information Processing Systems - Workshops (2023)
52. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Deep graph contrastive representation learning. In: International Conference on Machine Learning - Workshops (2020)
53. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Graph contrastive learning with adaptive augmentation. In: Proceedings of the Web Conference (2021)
54. Zhuo, J., Lu, Y., Ning, H., Fu, K., Niu, B., He, D., Wang, C., Guo, Y., Wang, Z., Cao, X., et al.: Unified graph augmentations for generalized contrastive learning on graphs. In: Advances in Neural Information Processing Systems (2024)

A Graph Convolutional Network

The graph encoder uses two graph convolution layers, which are mathematically represented as follows:

$$\mathbf{H}_1 = \sigma\left(\left(\mathbf{D}^{-\frac{1}{2}}\left(\mathbf{A}+\mathbf{I}\right)\mathbf{D}^{-\frac{1}{2}}\mathbf{X}\Theta_1\right)\right), \quad \mathbf{H}_2 = \sigma\left(\mathbf{D}^{-\frac{1}{2}}\left(\mathbf{A}+\mathbf{I}\right)\mathbf{D}^{-\frac{1}{2}}\mathbf{H}_1\Theta_2\right). \tag{14}$$

B Details of the KL Divergence in Subgraph Gaussian Embedding

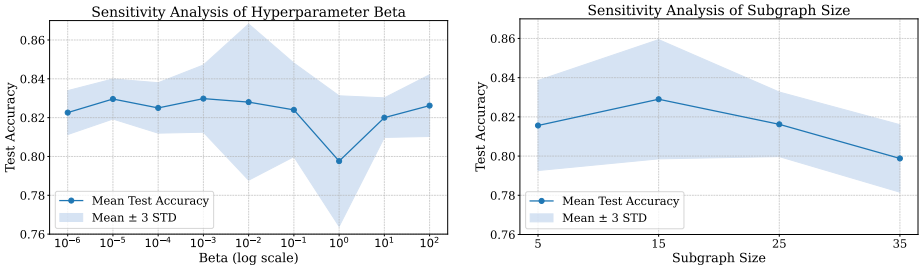
The KL divergence between these two distributions has a well-known closed-form expression. In our setting, we write [27]:

$$\mathrm{KL}\big(q(\tilde{\mathbf{X}}|\mathbf{X},\mathbf{A}) \,\big\|\, p(\tilde{\mathbf{X}})\big) = \frac{1}{2|\mathcal{P}|}\sum_{i\in\mathcal{P}}\sum_{j=1}^{d}\Big(\mu_{ij}^2 + \sigma_{ij}^2 - 1 - 2\,\log\sigma_{ij}\Big), \tag{15}$$

where μ_{ij} and σ_{ij} represent the j-th components of the latent mean and latent standard deviation for node i. The set $\mathcal{P}$ indexes the nodes in the induced subgraphs under consideration, and d is the dimensionality of the latent space.

C Sensitivity Analysis

To investigate the impact of the regularization constraint on our method, experiments were conducted on the Cora dataset. The influence of regularization



(a) Sensitivity analysis of hyperparameter beta β.

(b) Sensitivity analysis of subgraph size k^i.

Fig. 4: The plot displays the mean test accuracy (solid blue line) along with a shaded confidence region representing the mean ± 3 standard deviations. The analysis illustrates the sensitivity of test accuracy to variations in hyperparameter beta and subgraph sizes.

within the loss function was controlled by varying the hyperparameter β, which ranged from 10^{-6} to 10^{2}. The results, as illustrated in Figure 4a, indicate sensitivity to changes in β. Specifically, we observe that values of β greater than or equal to 10^{-5} have a pronounced effect on the model's performance. Optimal results on the Cora dataset are achieved when β was set within 10^{-3}.

To evaluate the sensitivity of SubGEC to the subgraph size hyperparameter, we conducted a sensitivity analysis on the Cora dataset using subgraph sizes $k = 5, 15, 25$, and 35. As shown in Figure 4b, the model exhibits robust performance across a wide range of subgraph sizes, with competitive mean test accuracy and low variability observed for $k = 5$ to 25. While $k = 15$ achieves marginally higher accuracy, the minimal differences in performance across this range suggest that the model is not overly sensitive to precise subgraph size selections. A gradual decline in performance at $k = 35$ highlights the upper bound of robustness, likely due to increased noise from redundant structural information.