Khan-GCL: Kolmogorov–Arnold Network Based Graph Contrastive Learning with Hard Negatives

Zihu Wang, Boxun Xu, Hejia Geng, Peng Li

University of California, Santa Barbara
{zihu_wang, boxunxu, hejia, lip}@ucsb.edu

Abstract

Graph contrastive learning (GCL) has demonstrated great promise for learning generalizable graph representations from unlabeled data. However, conventional GCL approaches face two critical limitations: (1) the restricted expressive capacity of multilayer perceptron (MLP) based encoders, and (2) suboptimal negative samples that are either generated from random augmentations—failing to provide effective 'hard negatives'—or hard negatives crafted without addressing the semantic distinctions crucial for discriminating graph data. To this end, we propose **Khan-GCL**, a novel framework that integrates the Kolmogorov–Arnold Network (KAN) into the GCL encoder architecture, substantially enhancing its representational capacity. Furthermore, we exploit the rich information embedded within KAN coefficient parameters to develop two novel critical feature identification techniques that enable the generation of semantically meaningful hard negative samples for each graph representation. These strategically constructed hard negatives guide the encoder to learn more discriminative features by emphasizing critical semantic differences between graphs. Extensive experiments demonstrate that our approach achieves state-of-the-art performance compared to existing GCL methods across a variety of datasets and tasks.

Code — https://github.com/zihuwang97/KhanGCL
Extended version — https://arxiv.org/pdf/2505.15103

Introduction

Graph Neural Networks (GNNs) are a class of machine learning models designed to learn from graph-structured data and are critical for tasks such as social network analysis, molecular property prediction, and recommendation systems. Integrating self-supervised contrastive learning (CL) that has gained popularity across a variety of domains (Oord, Li, and Vinyals 2018; Chen et al. 2020; Gao, Yao, and Chen 2021; Radford et al. 2021; Wang, Somayaji, and Li 2024; Wang et al. 2023a,b, 2024b) into graph learning has given rise to Graph Contrastive Learning (GCL), enabling pre-training GNN encoders from unlabeled graph data (Veličković et al. 2018; You et al. 2020; Zhu et al. 2020; You et al. 2021).

However, how to train good GCL models for real-world applications where labeled graphs are unavailable faces two challenges. First, existing GCL models employ Multilayer Perceptron (MLP) encoders while facing a dilemma: shallow MLPs limit the generalization ability of the encoder (Zhang et al. 2024) while deep MLPs can easily overfit (Chen et al. 2022; Rong et al. 2019). In addition, deep GNN encoders can overcompress, distort, or homogenize node features, making node representations indistinguishable from each other and leading to performance degradations (Li, Han, and Wu 2018). These difficulties have rendered use of MLP encoders with a limited depth. But in general, while being critical, striking a good balance between expressiveness and need for mitigating deep GNNs' inherent limitations is difficult.

Second, the performance of GCL heavily relies on the construction of augmented graph data pairs. Positive pairs consist of views derived from the same graph using augmentations (You et al. 2020, 2021), which help the encoder learn semantically similar graph features. Conversely, negative pairs, comprising different graphs, provide crucial information about semantic differences and thus encourage the learning of discriminative features (You et al. 2020, 2021; Xia et al. 2021). In the CL literature, *hard negatives* refer to samples from different classes that share similar latent semantic features with a target data point. Recent studies (Kalantidis et al. 2020; Xia et al. 2021; Luo et al. 2023; Wang et al. 2024a) demonstrate that incorporating such hard negatives significantly improves the encoder's performance on downstream tasks by making contrastive loss minimization more challenging. However, generating high-quality hard negatives remains a non-trivial task. The methods in (Chen et al. 2020; Cui et al. 2021) enlarge the training batch size to include more negatives, but without guaranteeing inclusion of more hard negatives, this can lead to performance degradation(Kalantidis et al. 2020). Additionally, the absence of labels in unsupervised pre-training renders the introduction of 'false negatives', formed by pairs of samples belonging to the same class (Kalantidis et al. 2020; Xia et al. 2021). Adversarial approaches generate negatives without explicitly identifying which latent features are most crucial to discriminate negative pairs (Hu et al. 2021; Luo et al. 2023; Zhang, Yang, and Shi 2024; Wang et al. 2024a). Therefore, more effective methods are desired for generating hard negative pairs to improve the performance of GCL.

We believe that tackling the challenges brought by lack of labeled graph data and the inherent GNN problems in real-

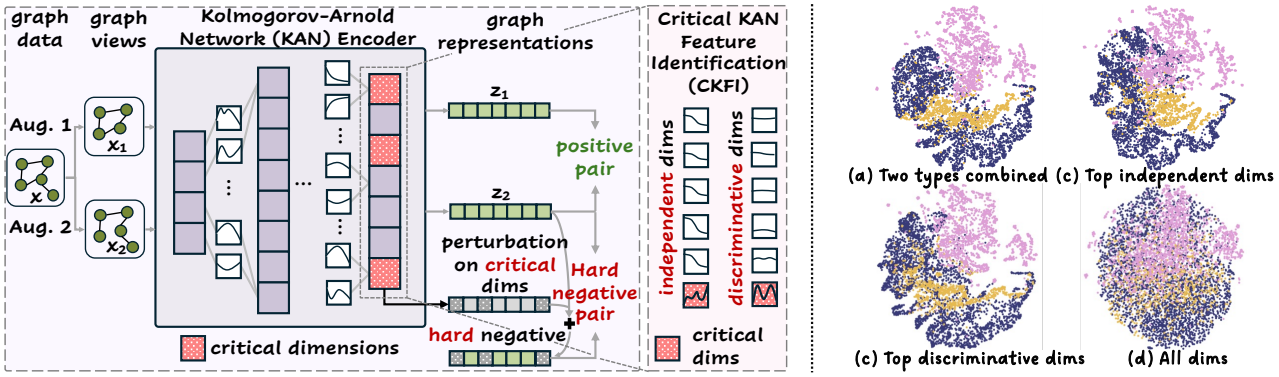

Figure 1: **Left**: Overview of the Khan-GCL framework. The encoder utilizes KAN to enhance expressive power and interpretability. Leveraging the KAN architecture, we introduce two critical dimension identification techniques. By applying small perturbations to these identified dimensions, we generate hard negative samples for each graph, thereby improving the performance of this GCL approach. **Right**: UMAP(2018) visualization of the pre-trained KAN encoder's output feature vectors on the COLLAB dataset(Morris et al. 2020). Points are colored by class (three classes). We use CKFI to calculate features' discriminative and independent scores using Equation 6 and 8. In (d), all dimensions yield poor separation. In (b) and (c), removing less critical dimensions, the top 25% most independent and discriminative features improve data separation. In (a), combining both types of feature dimensions (removing duplicates) achieves the best separation.

world applications requires advances in both GCL model architecture and data augmentation. To this end, we propose **Khan-GCL**: **K**AN-based **ha**rd **n**egative generation for **GCL**. In terms of model architecture, we replace typical MLP encoders with Kolmogorov-Arnold Network (KAN) (Liu et al. 2024). By integrating the Kolmogorov-Arnold representation theorem into modern neural networks, KANs introduce learnable non-linear activation functions in the network, replacing fixed activation functions in MLPs (Hornik, Stinchcombe, and White 1989; Cybenko 1989). KANs impose a localized structure on the trainable functions through their spline-based kernel activations. This acts as a form of regularization, guiding the model to learn smoother mappings that generalize better. This increases representational power without requiring a deeper network, which is critical for avoiding overfitting when data is scarce. As a result, KAN-based encoders strike a better balance between expensiveness and risks of inherent issues in deep GNNs.

In terms of data augmentation, we develop Critical KAN Feature Identification (**CKFI**), for generating hard negatives in the representation space of KAN encoders. By exploiting the nature of the B-spline coefficients, **CKFI** identifies two types of critical features—*discriminative* and *independent* features, highlighting the most sensitive and distinctive dimensions of the underlying graph structure. Applying small perturbations to these critical features changes the essential semantics of the graph while keeping it similar to the original graph, hence generates hard negative pairs. Such high-quality hard negatives improve the encoder's ability to discriminate subtle but crucial semantics in downstream tasks.

Our main contributions of this paper include:

1. We propose **Khan-GCL**, the first graph contrastive learning framework that integrates Kolmogorov-Arnold Network (KAN) encoders into contrastive learning, increasing the expressive power of GNNs.
2. We introduce **CKFI**, a novel approach for identifying two types of critical KAN output features, which are most independent and most discriminative of varying underlying graph structures, by exploiting the global nature of the learned B-spline coefficients from KAN encoders.
3. We present a new method that minimally perturbs the most critical output features of each KAN encoder to generate semantically meaningful hard negatives, thereby enhancing the effectiveness of graph contrastive learning.

Experiments across various biochemical and social media datasets demonstrate that our method achieves state-of-the-art performance on different tasks.

Related Works

Graph Contrastive Learning (GCL)

Graph Contrastive Learning (GCL) aims to learn powerful representations from unlabeled graph data (Veličković et al. 2018; You et al. 2020; Zhu et al. 2020; You et al. 2021). Typically, GCL employs random data augmentation strategies to generate diverse views of graphs to form positive and negative pairs(You et al. 2020). Subsequent research has introduced automated (You et al. 2021), domain-knowledge informed (Zhu et al. 2021; Wang et al. 2021), and saliency-guided (Li et al. 2025; Liu et al. 2021; Li et al. 2022) augmentation approaches to further improve representation quality.

Recent studies in CL (Kalantidis et al. 2020; Xia et al. 2021) emphasize the significance of hard negatives, demonstrating their effectiveness in enhancing downstream task performance. While prior research (Chen et al. 2020; Cui et al. 2021) suggests that enlarging training batch size to include more negative samples can improve feature discrimination, merely increasing the number of negatives does not inherently yield harder negative samples. In fact, continually increasing batch size can lead to performance degradation

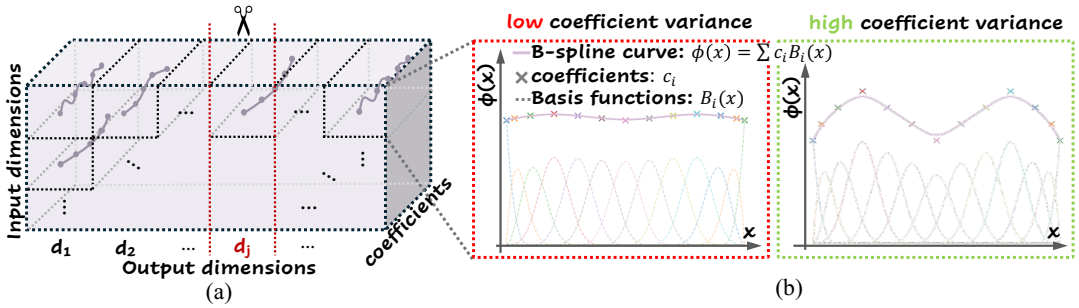

Figure 2: Illustration of a KAN layer and our two proposed critical feature identification techniques. (a) An output dimension is deemed *independent* when its removal from the coefficient tensor prevents accurate reconstruction of the original tensor (i.e., results in great reconstruction error). (b) Output dimensions comprising B-splines with high coefficient variance are considered *discriminative*, as larger coefficient variance typically corresponds to greater functional variance.

in contrastive learning (Kalantidis et al. 2020). To explicitly introduce hard negatives, (Kalantidis et al. 2020; Xia et al. 2021) propose ranking existing negatives in a mini-batch based on their similarity and mixing the hardest examples to produce hard negatives. However, the absence of labels in unsupervised pre-training may introduce false negatives. More recent approaches frame hard negative generation in GCL as an adversarial process (Hu et al. 2021; Luo et al. 2023; Zhang, Yang, and Shi 2024; Wang et al. 2024a), wherein a negative generator attempts to maximize the contrastive loss while the encoder aims to minimize it. However, current adversarial approaches primarily focus on bringing negative pairs closer without explicitly identifying critical latent dimensions responsible for semantic differences.

Kolmogorov-Arnold Networks in Graph Learning

Kolmogorov-Arnold Network (KAN) (Liu et al. 2024) is a novel neural network architecture demonstrating improved generalization ability and interpretability over MLPs. Many recent studies (Kiamari, Kiamari, and Krishnamachari 2024; Zhang and Zhang 2024; Bresson et al. 2025; Li et al. 2024b) have proposed KAN-based GNN architectures by directly replacing MLPs in conventional models with KAN. These studies reveal that KAN effectively enhances the expressiveness of traditional GNNs while mitigating the over-squashing problem inherent in MLP-based architectures. Furthermore, several studies have demonstrated KAN's superior generalizability over MLPs in specialized domains, including drug discovery (Ahmed and Sifat 2024), molecular property prediction (Li et al. 2024a), recommendation systems (Xu et al. 2024), and smart grid intrusion detection (Wu et al. 2025).

Beyond straightforward MLP substitution, KAA (Fang et al. 2025) embeds KANs into the attention scoring functions of GAT and Transformer-based models. KA-GAT (Chen et al. 2025) combines KAN-based feature decomposition with multi-head attention and graph convolutions, enhancing the model's capacity in high-dimensional graph data.

Preliminary

Graph Contrastive Learning aims to train an encoder $f(\cdot)$ which maps graph data $\mathbf{x} \in \mathbb{R}^m$ to representations $\mathbf{z} \in \mathbb{R}^n$. In pre-training, a projection head $h(\cdot)$ is often employed to project the representations to $v \in \mathbb{R}^p$. GCL typically applies two random augmentation functions $A_1(\cdot)$ and $A_2(\cdot)$, sampled from a set $\mathcal{A}$ of augmentations, to produce two views of each graph from a batch $\mathcal{B}_o = \{\mathbf{x}_i^o\}_{i=1}^N$ to get a batch of augmented graphs $\mathcal{B} = \{\mathbf{x}_i\}_{i=1}^{2N}$, where $\mathbf{x}_{2i} = A_1(\mathbf{x}_i^0)$, $\mathbf{x}_{2i+1} = A_2(\mathbf{x}_i^0)$. These views are then encoded and projected to $\mathbb{R}^p$, i.e., $\mathbf{z}_i = f(\mathbf{x}_i)$ and $\mathbf{v}_i = h(\mathbf{z}_i)$. A contrastive loss (Chen et al. 2020; You et al. 2020) is applied to encourage similarity between positive pairs and dissimilarity between negative pairs:

$$\mathcal{L}_{CL} = \frac{1}{2N}\sum_{i=1}^{N} -[\log \frac{\exp(\text{sim}(\mathbf{v}_{2i}, \mathbf{v}_{2i+1})/\tau)}{\sum_{j\neq 2i}\exp(\text{sim}(\mathbf{v}_{2i}, \mathbf{v}_j)/\tau)} + \log \frac{\exp(\text{sim}(\mathbf{v}_{2i+1}, \mathbf{v}_{2i})/\tau)}{\sum_{j\neq 2i+1}\exp(\text{sim}(\mathbf{v}_{2i+1}, \mathbf{v}_j)/\tau)}] \tag{1}$$

Here $\text{sim}(\cdot,\cdot)$ calculates the cosine similarity between two vectors. τ is a temperature hyperparameter. Although various GCL methods have been developed, their contrastive losses are defined similarly.

Kolmogorov-Arnold Networks integrate the Kolmogorov-Arnold representation theorem into modern neural networks. A KAN layer $\mathbf{\Phi}$, which maps input data from $\mathbb{R}^{d_{in}}$ to $\mathbb{R}^{d_{out}}$, is defined as:

$$x_j^{out} = \sum_{i=1}^{d_{in}} \phi_{i,j}(x_i^{in}) \qquad \forall j \in \{1, 2, \ldots, d_{out}\} \tag{2}$$

x_i^{in} and x_j^{out} denote the input and output components at dimensions i and j, respectively. Each univariate function $\phi_{i,j}$ represents a learnable non-linear function associated with the connection from the i_{th} input dimension to the j_{th} output dimension. These functions are usually parameterized using B-splines (Liu et al. 2024), such that:

$$\phi_{i,j}(\cdot) = \sum_k c_{ijk} B_{ijk}(\cdot) \tag{3}$$

$B_{ijk}(\cdot)$ denotes the B-spline basis functions for $\phi_{i,j}(\cdot)$, and c_{ijk} represents their corresponding trainable coefficients. All coefficients at a KAN layer can thus be denoted as $\mathcal{C} = \{c_{ijk}\} \in \mathbb{R}^{d_{in}\times d_{out}\times d_c}$, where d_c is the number of coefficients used to define each B-spline function.

Method

Overview

Figure 1 (left) illustrates **Khan-GCL**, the first GCL framework with a KAN-based encoder. By parameterizing non-linearity with trainable basis function coefficients, KANs offer improved generalizability and interpretability over conventional MLPs. Leveraging the rich non-linear information in KAN coefficients, we introduce **CKFI** to identify two types of critical features. Small perturbations applied to these features generate hard negatives that alter sample semantics while preserving structural similarity. Incorporating these hard negatives during pre-training guides the encoder to learn more discriminative graph features.

Critical KAN Feature Identification (CKFI)

Independent Dimensions in KANs According to Equation 2, each latent dimension in a KAN layer combines a unique set of non-linear functions, capturing distinct non-linear features from the input. However, as each B-spline function is a linear combination of basis functions $B_{ijk}(\cdot)$ (Equation 3), any output dimension whose coefficients are linearly dependent on those of other dimensions can be expressed as a linear combination of them, indicating redundancy.

Proposition 1 (KAN layer output dependency)**.** *For a KAN layer with coefficients $\mathcal{C} = \{c_{ijk}\} \in \mathbb{R}^{d_{in} \times d_{out} \times d_c}$, we denote the slice corresponding to output dimension d by $\mathcal{C}_{:,d,:} \in \mathbb{R}^{d_{in} \times d_c}$. If $\mathcal{C}_{:,d,:}$ is a linear combination of $\mathcal{C}_{:,d_1,:}, \mathcal{C}_{:,d_2,:}, \ldots, \mathcal{C}_{:,d_n,:}$, the dimension d's output feature can be expressed as a linear combination of dimensions $d_1, d_2, \ldots, d_n$.*

Thus, conversely, we define an *independent dimension* as one whose coefficients cannot be expressed as linear combinations of coefficients from other dimensions, thus encoding truly unique features globally from the entire input domain.

To identify independent dimensions in a KAN layer, as illustrated in Figure 2(a), we attempt to reconstruct the original coefficient tensor $\mathcal{C}$ of the layer after removing coefficients from each output dimension. Specifically, given the coefficients $\mathcal{C} = [c_{ijk}] \in \mathbb{R}^{d_{in} \times d_{out} \times d_c}$ from a KAN layer, we perform higher-order singular value decomposition (HOSVD) with each output dimension $d_j, 1 \leq j \leq d_{out}$ removed as follows:

$$\mathcal{C}^{(-j)} \approx \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \tag{4}$$

Here $\mathcal{C}^{(-j)} \in \mathbb{R}^{d_{in} \times (d_{out}-1) \times d_c}$ denotes the tensor obtained by removing the j_{th} mode-2 slice from $\mathcal{C}$. $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ is the core tensor containing singular values of $\mathcal{C}^{(-j)}$, and $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}$ are orthogonal bases for each mode of $\mathcal{C}^{(-j)}$. The notation $\times_n$ denotes the mode-n tensor product.

Then we reconstruct the coefficients $\mathcal{C}$ using $\mathcal{C}^{(-j)}$. First, we project the removed j_{th} mode-2 slice back to the subspace spanned by $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(3)}$ to get $\tilde{\mathbf{M}}_j^{(2)}$. Integrating $\tilde{\mathbf{M}}_j^{(2)}$ into $\mathbf{U}^{(2)} \in \mathbb{R}^{r_2 \times (d_{out}-1)}$, we get $\tilde{\mathbf{U}}_j^{(2)} \in \mathbb{R}^{r_2 \times d_{out}}$, the approximated basis $\mathbf{U}_j^{(2)}$ including the j_{th} dimension. Subsequently, we reconstruct $\mathcal{C}$ as follows:

$$\tilde{\mathcal{C}}_j = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \tilde{\mathbf{U}}_j^{(2)} \times_3 \mathbf{U}^{(3)} \tag{5}$$

where $\tilde{\mathcal{C}}_j$ is the reconstruction with the j_{th} mode-2 slice removed from $\mathcal{C}$. The reconstruction error can be calculated using the Frobenius norm:

$$\delta_j = \|\tilde{\mathcal{C}}_j - \mathcal{C}\|_F \tag{6}$$

A larger reconstruction error δ_j indicates that the j_{th} output dimension encodes more unique features, as it becomes difficult to be accurately reconstructed when excluded from the tensor decomposition.

Discriminative Dimensions in KANs In this section, we focus on *discriminative dimensions*, providing an orthogonal perspective to the aforementioned independent dimensions. In downstream discrimination tasks, latent dimensions exhibiting larger output variance are preferred since they effectively separate different data points.

To implement discriminative feature identification in practice, one can naively sample mini-batches from a dataset and calculate the variance at each dimension. However, estimating variance from randomly sampled mini-batches can introduce bias and additional computational overhead. Instead, we examine the coefficients of KAN layers, which provide a 'global' view of the distribution of the underlying features. As illustrated in Figure 2(b), the shape of a B-spline curve closely aligns with the pattern of its coefficients. To formally establish this observation, we present the following proposition, establishing an upper bound on the variance of a uniform B-spline function based on its coefficients' variance.

Proposition 2 (Variance Upper Bound of a Uniform B-spline Function)**.** *$\phi(x) = \sum_k c_k B_k(x)$ is a uniform B-spline function defined over the interval $[a, b]$. $\{B_k(x)\}_{k=1}^{d_c}$ denotes the set of uniform B-spline basis functions and $\{c_k\}_{k=1}^{d_c}$ are the corresponding coefficients. The variance of $\phi(x)$ over its input domain satisfies the following inequality:*

$$\mathrm{Var}[\phi(x)] = \int_a^b (\phi(x) - \mu_\phi)^2 dx \leq M(0) \cdot \sigma_c^2, \tag{7}$$

where $\mu_\phi = \frac{1}{b-a} \int_a^b \phi(x) dx$ is the mean of $\phi(x)$, $M(0) = \int [B_k(x)]^2 dx$ is identical for all $B_k(x)$ for a uniform B-spline, and $\sigma_c^2 = \frac{1}{d_c} \sum_{k=1}^{d_c} (c_k - \mu_c)^2$ represents the variance of the coefficients $\{c_k\}_{k=1}^{d_c}$ with $\mu_c = \frac{1}{d_c} \sum_{k=1}^{d_c} c_k$ being the mean.

Equation (7) implies that the variance of the output of a B-spline function over its input domain is bounded by the variance of its coefficients. Consequently, as illustrated in Figure 2(b), we exploit the variance of coefficients in each feature dimension to assess the discriminative power of that dimension. Specifically, in a KAN layer whose coefficients are denoted by $\mathcal{C} = [c_{ijk}] \in \mathbb{R}^{d_{in} \times d_{out} \times d_c}$, we quantify the discriminative capability of an output dimension j using the average variance across all B-spline functions associated with that dimension as follows:

$$\rho_j = \frac{1}{d_{in}} \cdot \sum_{i=1}^{d_{in}} \sigma_{c_{ij}}^2 \tag{8}$$

where $\mu_{c_{ij}} = \frac{1}{d_c} \sum_{k=1}^{d_c} c_{ijk}$, and $\sigma_{c_{ij}}^2 = \frac{1}{d_c} \sum_{k=1}^{d_c} (c_{ijk} - \mu_{c_{ij}})^2$ denoting the variance of coefficients for the B-spline function $\phi_{i,j}(\cdot)$ linking the i^{th} input dimension to the j^{th} output dimension. A dimension j with a large ρ_j is considered more discriminative w.r.t. the input data.

Datasets	BBBP	Tox21	ToxCast	SIDER	ClinTox	MUV	HIV	BACE	AVG
AttrMasking(Hu et al. 2019)	64.3±2.8	76.7±0.4	64.2±0.5	61.0±0.7	71.8±4.1	74.7±1.4	77.2±1.1	79.3±1.6	71.1
GraphCL(You et al. 2020)	69.7±0.7	73.9±0.7	62.4±0.6	60.5±0.9	76.0±2.7	69.8±2.7	78.5±1.2	75.4±1.4	70.8
JOAOv2(You et al. 2021)	71.4±0.9	74.3±0.6	63.2±0.5	60.5±0.5	81.0±1.6	73.7±1.0	77.5±1.2	75.5±1.3	72.1
RGCL(Li et al. 2022)	71.4±0.7	75.2±0.3	63.3±0.2	61.4±0.6	83.4±0.9	76.7±1.0	77.9±0.8	76.0±0.8	73.2
GraphACL(Luo et al. 2023)	73.3±0.5	76.2±0.6	64.1±0.4	62.6±0.6	**85.0±1.6**	76.9±1.2	78.9±0.7	80.1±1.2	74.6
DRGCL(Ji et al. 2024)	71.2±0.5	74.7±0.5	64.0±0.5	61.1±0.8	78.2±1.5	73.8±1.1	78.6±1.0	78.2±1.0	72.5
CI-GCL(Tan et al. 2024)	**74.4±1.9**	77.3±0.9	65.4±1.5	64.7±0.3	80.5±1.3	76.5±0.9	80.5±1.3	**84.4±0.9**	75.4
Khan-GCL	73.5±0.6	**78.3±0.3**	**66.3±0.3**	**65.0±0.9**	84.3±1.2	**77.5±0.4**	**80.7±0.6**	80.9±1.0	**75.8**

Table 1: Transfer learning performance (ROC-AUC scores in %) for graph classification across 8 datasets. Results for benchmark methods are reported from their respective original publications.

Hard Negatives in Khan-GCL

Prior research (Kalantidis et al. 2020; Xia et al. 2021) suggests that in contrastive learning, the most effective hard negatives for a graph satisfy two key criteria: (a) their key identity is different from the original graph, and (b) they maintain high semantic similarity to the original graph. Therefore, to generate an effective hard negative for a graph, we aim to produce a variant with **minimal deviation from the original** while strategically **distorting its key characteristics**. As shown in Figure 1 (right), two types of CKFI dimensions include the most important features to recognize data from different classes. Applying small perturbations to them can thus effectively change the identity of a feature vector.

We propose applying small perturbations to graph representations from the encoder's last layer to generate hard negatives in the output space of the encoder. Specifically, we define these perturbations as follows:

$$\begin{aligned}\mathbf{p}^{\delta} &= \{p_i^{\delta} = \alpha_i \cdot u_i^{\delta} : u_i^{\delta} \sim \mathcal{N}(\epsilon_{\delta} \cdot \delta_i, \sigma_{\delta}^2), \alpha_i \sim \mathrm{Rad}\} \\ \mathbf{p}^{\rho} &= \{p_i^{\rho} = \alpha_i \cdot u_i^{\rho} : u_i^{\rho} \sim \mathcal{N}(\epsilon_{\rho} \cdot \rho_i, \sigma_{\rho}^2), \alpha_i \sim \mathrm{Rad}\}\end{aligned} \tag{9}$$

Here $\epsilon_{\delta} > 0$ and $\epsilon_{\rho} > 0$ are hyperparameters scaling $\boldsymbol{\delta} = \{\delta_i\}$ and $\boldsymbol{\rho} = \{\rho_i\}$ computed by the proposed CKFI method for feature i per Equations (6) and (8), respectively. σ_{δ}^2 and σ_{ρ}^2 represent the variance hyperparameters of the Gaussian distributions. With these perturbations, dimensions that are highly discriminative or independent (i.e., those with large ρ_i or δ_i values) receive perturbations from Gaussian distributions with larger means. Since $\epsilon_{\rho} \cdot \rho_i > 0$ and $\epsilon_{\delta} \cdot \delta_i > 0$ for all dimensions i, we introduce α_i sampled from the Rademacher distribution Rad to ensure that perturbations p_i^{ρ} and p_i^{δ} are approximately equally likely to be positive or negative. With these perturbations critical dimensions receive more substantial perturbations on average.

During training, with a mini-batch of N graphs, the augmented graph data is denoted by $\mathcal{B} = \{\mathbf{x}_j\}_{j=1}^{2N}$ of size $2N$, and their latent representations are $\mathcal{B}_z = \{\mathbf{z}_j\}_{j=1}^{2N}$. For each representation $\mathbf{z}_j$ in $\mathcal{B}_z$, we sample perturbation vectors $\mathbf{p}_j^{\rho}$ and $\mathbf{p}_j^{\delta}$ and produce the hard negative of $\mathbf{z}_j$ as:

$$\mathbf{z}_j^{hard} = \mathbf{z}_j + \mathbf{p}_j^{\rho} + \mathbf{p}_j^{\delta} \tag{10}$$

The generated hard negative $\mathbf{z}_j^{hard}$ is then projected to $\mathbf{v}_j^{hard}$ by the projection head and utilized in our proposed hard negative loss:

$$\mathcal{L}_{HN} = \frac{1}{2N} \sum_{j=1}^{2N} \log[\exp(\mathrm{sim}(\mathbf{v}_j, \mathrm{sg}(\mathbf{v}_j^{hard})))] \tag{11}$$

To prevent model collapse, we apply a stop-gradient operator $\mathrm{sg}(\cdot)$ to hard negatives $\mathbf{v}_j^{hard}$. Finally, we write the overall training loss $\mathcal{L}_{Khan}$ of Khan-GCL as:

$$\mathcal{L}_{Khan} = \mathcal{L}_{CL} + \mathcal{L}_{HN} \tag{12}$$

Experiments

In this section, we conduct comprehensive experiments across diverse datasets and tasks to demonstrate the efficacy of our approach. Furthermore, we conduct extensive ablation studies to provide deeper insights into the mechanisms underlying our proposed method. In all tables, **bold values** denote the best performance on the corresponding dataset, while underlined values indicate the second-best performance. All experiments are run on a single NVIDIA A100 GPU.

Model architecture and hyperparameters. For fair comparison, we follow the general contrastive learning hyperparameter settings of (You et al. 2020). In our KAN-based encoder implementation, we systematically replace all MLPs in the backbone architectures of (You et al. 2020) with Cubic KAN layers (utilizing 3rd order B-spline functions) while maintaining identical input, output, and hidden dimensions.

Datasets. We conduct experiments on Zinc-2M (Sterling and Irwin 2015), 8 biochemical datasets from (Wu et al. 2018), 8 diverse biochemical/social network datasets from the TU-datasets collection (Morris et al. 2020), and MNIST-superpixel (Monti et al. 2017).

Main Results

Transfer learning is a widely adopted evaluation protocol for assessing the generalizability and transferability of representations learned by GCL methods. We pre-train our backbone encoder on the large-scale molecular dataset Zinc-2M (Sterling and Irwin 2015) using the proposed Khan-GCL framework, then finetune the pre-trained encoder on 8 biochemical datasets for graph classification tasks. Table 1 shows the graph classification accuracy across these datasets.

Datasets	DD	MUTAG	NCI1	PROTEINS	COLLAB	RDT-B	RDT-M5K	IMDB-B	AVG
InfoGraph (Sun et al. 2019)	72.9±1.8	89.0±1.1	76.2±1.0	74.4±0.3	70.1±1.1	82.5±1.4	53.5±1.0	73.0±0.9	74.0
GraphCL(You et al. 2020)	78.6±0.4	86.8±1.3	77.9±0.4	74.4±0.5	71.4±1.1	89.5±0.8	56.0±0.3	71.1±0.4	75.7
JOAOv2(You et al. 2021)	77.4±1.1	87.7±0.8	78.4±0.5	74.1±1.1	69.3±0.3	86.4±1.5	56.0±0.3	70.1±0.3	74.9
AD-GCL (Suresh et al. 2021)	75.8±0.9	88.7±1.9	73.9±0.8	73.3±0.5	72.0±0.6	90.1±0.9	54.3±0.3	70.2±0.7	74.8
RGCL(Li et al. 2022)	78.9±0.5	87.7±1.0	78.1±1.1	75.0±0.4	71.0±0.7	90.3±0.6	56.4±0.4	71.9±0.9	76.2
DRGCL(Ji et al. 2024)	78.4±0.7	89.5±0.6	78.7±0.4	75.2±0.6	70.6±0.8	90.8±0.3	56.3±0.2	72.0±0.5	76.4
TopoGCL(2024)	79.1±0.3	90.1±0.9	**81.3±0.3**	**77.3±0.9**	-	90.4±0.5	-	74.7±0.3	-
CI-GCL(Tan et al. 2024)	79.6±0.3	89.7±0.9	80.5±0.5	76.5±0.1	74.4±0.6	90.8±0.5	56.6±0.3	73.8±0.8	77.7
Khan-GCL(Ours)	**80.6±0.7**	**91.4±1.1**	80.8±0.9	76.9±0.8	**75.2±0.3**	**92.2±0.3**	**56.9±0.5**	**75.0±0.4**	**78.6**

Table 2: Unsupervised learning performance (accuracy in %) for graph classification on TU-datasets. Results for benchmark methods are quoted from their original publications, except for AD-GCL and InfoGraph, which are reported from (Li et al. 2022).

Datasets	DD	MUTAG	NCI1	PROTEINS	COLLAB	RDT-B	RDT-M5K	IMDB-B
AFANS(Wang et al. 2024a)	-	90.0±1.0	80.4±0.5	75.4±0.5	74.7±0.5	91.1±0.1	-	-
ANGCL(Zhang, Yang, and Shi 2024)	78.8±0.9	**92.3±0.7**	**81.0±0.3**	75.9±0.4	72.0±0.7	90.8±0.7	56.5±0.3	71.8±0.6
GraphACL(Luo et al. 2023)	79.3±0.4	90.2±0.9	-	75.5±0.4	74.7±0.6	-	-	74.3±0.7
Khan-GCL(Ours)	**80.6±0.7**	91.4±1.1	80.8±0.9	**76.9±0.8**	**75.2±0.3**	**92.2±0.3**	**56.9±0.5**	**75.0±0.4**

Table 3: Unsupervised learning performance (accuracy in %) for graph classification on TU-datasets, comparing Khan-GCL against existing hard negative integrated GCL methods. Benchmark results are from their corresponding original publications.

Datasets	DD	MUTAG	NCI1	PROTEINS
GraphCL	78.6±0.4	86.8±1.3	77.9±0.4	74.4±0.5
GraphCL (KAN)	78.9±0.6	88.4±0.9	78.0±1.0	75.1±0.3
Ours	**80.6±0.7**	**91.4±1.1**	**80.8±0.9**	**76.9±0.8**
JOAOv2	77.4±1.1	87.7±0.8	78.4±0.5	74.1±1.1
JOAOv2 (KAN)	78.7±0.5	88.6±0.8	79.0±0.5	75.3±0.6
JOAOv2+Ours	**80.2±1.1**	**92.1±0.3**	**81.5±0.7**	**77.0±0.9**

Table 4: Ablation study results on the effectiveness of KAN in Khan-GCL. Unsupervised learning results (in %) for graph classification on TU-datasets are reported.

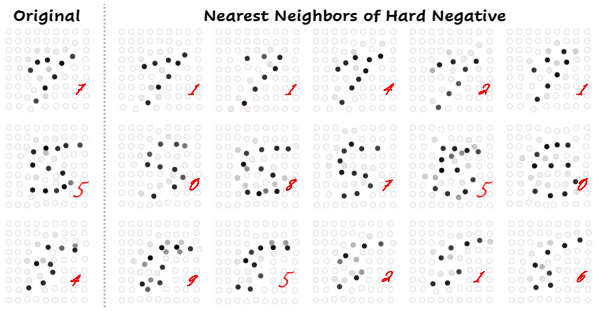

Figure 3: MNIST-superpixel graphs with their generated hard negatives' nearest neighbors in the latent space. Red numbers indicate ground truth labels. The nearest neighbors of a sample's hard negatives are typically similar to the sample while belonging to different classes.

Khan-GCL achieves the best overall results compared to all state-of-the-art methods. KAN's better generalization capability and the hard negative generation technique enhance the encoder's generalizability and ability to discriminate between subtle yet critical differences across graph structures.

Unsupervised learning aims to assess a GCL pre-training method's efficacy on diverse datasets. Following established protocols in (Sun et al. 2019; You et al. 2020), we pre-train our encoder on 8 datasets from TU-datasets (Morris et al. 2020). Subsequently, we employ an SVM classifier to evaluate the pre-trained encoder's feature representation quality.

Table 2 presents a comprehensive comparison between Khan-GCL and other state-of-the-art GCL approaches in unsupervised learning experiments. Khan-GCL achieves the best performance on 6 out of 8 datasets and yields the best overall results across all methods due to the powerful KAN encoder and effective generation of hard negatives. Furthermore, Table 3 compares Khan-GCL's effectiveness against existing hard negative integrated GCL methods, where our approach attains optimal results on 6 out of 8 datasets.

By targeting critical dimensions identified through CKFI, Khan-GCL emphasizes essential semantic features, which significantly enhance its classification capabilities compared to existing hard negative integrated methods.

Nearest neighbors retrieval of the generated hard negatives. To better elucidate the effectiveness of the generated hard negatives, we pre-train an encoder using Khan-GCL on MNIST-superpixel (Monti et al. 2017), where handwritten digits from the MNIST dataset (LeCun et al. 1998) are represented as graphs. During pre-training, we generate hard negatives for each graph and retrieve the nearest neighbors of each hard negative in the feature space across the entire dataset. Figure 3 illustrates sample digit graphs alongside the

Datasets	DD	MUTAG	NCI1	PROTEINS	COLLAB	RDT-B	RDT-M5K	IMDB-B	AVG
GraphCL (You et al. 2020)	78.6±0.4	86.8±1.3	77.9±0.4	74.4±0.5	71.4±1.1	89.5±0.8	56.0±0.3	71.1±0.4	75.7
w/o hard neg.	78.9±0.6	88.4±0.9	78.0±1.0	75.1±0.3	73.8±0.5	90.0±1.3	56.3±0.4	73.7±0.4	76.8
w/ rand-perturb	78.3±0.9	87.5±0.6	78.7±0.3	76.2±1.1	72.9±1.3	91.1±0.3	56.1±0.2	72.2±1.2	76.6
w/o d-dims	79.4±1.1	89.6±0.4	79.1±0.6	76.9±0.7	74.7±0.9	91.7±0.8	56.0±0.7	73.9±0.4	77.7
w/o i-dims	80.2±0.7	89.2±0.9	79.9±0.5	76.2±1.1	**75.5±0.5**	90.6±1.2	56.3±0.5	74.4±0.9	77.8
Khan-GCL	**80.6±0.7**	**91.4±1.1**	**80.8±0.9**	**76.9±0.8**	75.2±0.3	**92.2±0.3**	**56.9±0.5**	**75.0±0.4**	78.6

Table 5: Ablation study results for Khan-GCL. Unsupervised learning results (in %) for graph classification on TU-datasets are reported.

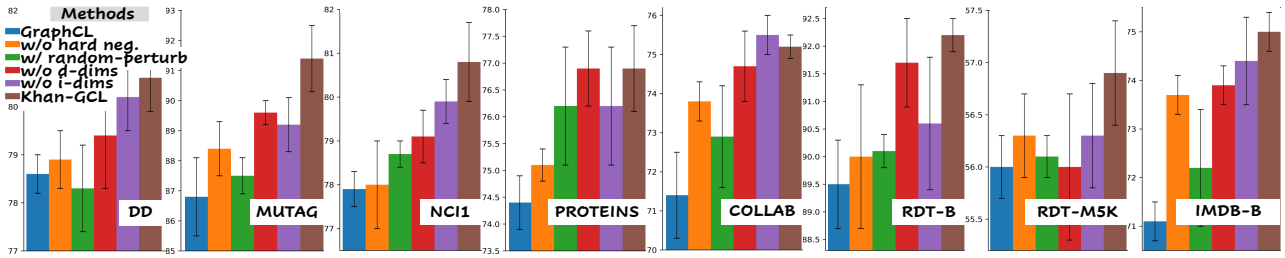

Figure 4: Ablation study results for Khan-GCL. Unsupervised learning results (in %) for graph classification on TU-datasets are shown.

five nearest neighbors of their corresponding hard negatives. The ground truth label for each graph is displayed in red at the bottom right corner. As shown in Figure 3, the nearest neighbors of a sample's hard negatives typically exhibit structural similarity to the original sample while belonging to different classes. This observation confirms that forming negative pairs between a sample and such generated hard negatives effectively guides the encoder to discriminate between semantically similar samples from different classes.

Ablation Study

Effectiveness of the KAN Encoder and Compatibility of Khan-GCL with Existing GCL Methods. We conduct targeted experiments to assess the KAN encoder's impact on Khan-GCL. Specifically, we evaluate Khan-GCL variants without CKFI and hard negative generation ('GraphCL (KAN)' and 'JOAOv2 (KAN)' in Table 4) against their baselines. We also introduce 'JOAOv2+Ours', which applies the full Khan-GCL framework to JOAOv2. Results show that KAN-enhanced variants outperform their counterparts even without hard negatives, demonstrating KAN's superior ability in modeling non-linearity. Adding hard negatives yields further gains, confirming its effectiveness and Khan-GCL's compatibility with diverse GCL methods.

Effectiveness of two feature identification techniques in hard negative generation. In this section, we evaluate the contributions of our two proposed critical feature identification techniques in hard negative generation. We introduce three variants of Khan-GCL: (1) 'w/o d-dims', where negatives are generated by perturbing only independent dimensions; (2) 'w/o i-dims', where perturbation is limited to discriminative dimensions; and (3) 'w/ rand-perturb', which generates negatives by applying random Gaussian noise across all dimensions. Figure 4 presents the performance of these configurations on unsupervised learning tasks. While 'w/ rand-perturb' yields performance improvements over 'w/o hard neg.' (a Khan-GCL variant with KAN encoder but without hard negative generation) on several datasets, it occasionally results in performance degradation. Both specialized perturbation approaches ('w/o d-dims' and 'w/o i-dims') outperform the baseline, demonstrating the effectiveness of targeted feature identification technique. Additionally, Khan-GCL, which integrates both proposed feature identification techniques in CKFI, achieves the most substantial improvement and the best overall results, confirming the complementary nature of our dual feature identification approach.

Conclusion

We propose **Khan-GCL**, the first KAN-based graph contrastive learning framework, which balances expressive power and risks of inherent issues of deep GNNs by using a KAN-based encoder. We also introduce **CKFI** to identify discriminative and independent features, enabling the generation of hard negatives through minimal perturbations. These hard negatives guide the encoder to learn critical semantics during contrastive pre-training. Extensive experiments on biochemical and social network datasets demonstrate that our approach significantly improves generalization and transferability, achieving the state-of-the-art performance. **Additional details and experimental results are provided in the Extended version.**

For future work, exploring feature perturbation and hard negative generation in intermediate layers of a KAN encoder is promising. Further, reducing the additional training cost introduced by KAN's spline computations through more efficient architectures is worth investigating.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No.1956313 and No.2334380.

References

Ahmed, T.; and Sifat, M. H. R. 2024. GraphKAN: Graph Kolmogorov Arnold Network for Small Molecule-Protein Interaction Predictions. In *ICML'24 Workshop ML for Life and Material Science: From Theory to Industry Applications*.

Bresson, R.; Nikolentzos, G.; Panagopoulos, G.; Chatzianastasis, M.; Pang, J.; and Vazirgiannis, M. 2025. KAGNNs: Kolmogorov-Arnold Networks meet Graph Learning. arXiv:2406.18380.

Chen, J.; Yuchi, X.; Yan, Z.; Dong, K.; and Li, H. 2025. KA-GAT: Kolmogorov–Arnold based Graph Attention Networks.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PmLR.

Chen, T.; Zhou, K.; Duan, K.; Zheng, W.; Wang, P.; Hu, X.; and Wang, Z. 2022. Bag of tricks for training deeper graph neural networks: A comprehensive benchmark study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 2769–2781.

Chen, Y.; Frias, J.; and Gel, Y. R. 2024. TopoGCL: Topological Graph Contrastive Learning. *arXiv preprint arXiv:2406.17251*.

Cui, G.; Du, Y.; Yang, C.; Zhou, J.; Xu, L.; Zhou, X.; Cheng, X.; and Liu, Z. 2021. Evaluating modules in graph contrastive learning. *arXiv preprint arXiv:2106.08171*.

Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4): 303–314.

Fang, T.; Gao, T.; Wang, C.; Shang, Y.; Chow, W.; Chen, L.; and Yang, Y. 2025. KAA: Kolmogorov-Arnold Attention for Enhancing Attentive Graph Neural Networks. arXiv:2501.13456.

Gao, T.; Yao, X.; and Chen, D. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Hornik, K.; Stinchcombe, M.; and White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5): 359–366.

Hu, Q.; Wang, X.; Hu, W.; and Qi, G.-J. 2021. Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1074–1083.

Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; and Leskovec, J. 2019. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*.

Ji, Q.; Li, J.; Hu, J.; Wang, R.; Zheng, C.; and Xu, F. 2024. Rethinking dimensional rationale in graph contrastive learning from causal perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12810–12820.

Kalantidis, Y.; Sariyildiz, M. B.; Pion, N.; Weinzaepfel, P.; and Larlus, D. 2020. Hard negative mixing for contrastive learning. *Advances in neural information processing systems*, 33: 21798–21809.

Kiamari, M.; Kiamari, M.; and Krishnamachari, B. 2024. GKAN: Graph Kolmogorov-Arnold Networks. arXiv:2406.06470.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, L.; Zhang, Y.; Wang, G.; and Xia, K. 2024a. KA-GNN: Kolmogorov-Arnold Graph Neural Networks for Molecular Property Prediction. arXiv:2410.11323.

Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Li, R.; Li, M.; Liu, W.; and Chen, H. 2024b. GNN-SKAN: Harnessing the Power of SwallowKAN to Advance Molecular Representation Learning with GNNs. arXiv:2408.01018.

Li, S.; Luo, Y.; Zhang, A.; Wang, X.; Li, L.; Zhou, J.; and Chua, T.-S. 2025. Self-attentive rationalization for interpretable graph contrastive learning. *ACM Transactions on Knowledge Discovery from Data*, 19(2): 1–21.

Li, S.; Wang, X.; Zhang, A.; He, X.; and Chua, T.-S. 2022. Let Invariant Rationale Discovery Inspire Graph Contrastive Learning. In *ICML*.

Liu, S.; Wang, H.; Liu, W.; Lasenby, J.; Guo, H.; and Tang, J. 2021. Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*.

Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T. Y.; and Tegmark, M. 2024. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*.

Luo, X.; Ju, W.; Gu, Y.; Mao, Z.; Liu, L.; Yuan, Y.; and Zhang, M. 2023. Self-supervised graph-level representation learning with adversarial contrastive learning. *ACM Transactions on Knowledge Discovery from Data*, 18(2): 1–23.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; and Bronstein, M. M. 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5115–5124.

Morris, C.; Kriege, N. M.; Bause, F.; Kersting, K.; Mutzel, P.; and Neumann, M. 2020. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmLR.

Rong, Y.; Huang, W.; Xu, T.; and Huang, J. 2019. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*.

Sterling, T.; and Irwin, J. J. 2015. ZINC 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11): 2324–2337.

Sun, F.-Y.; Hoffmann, J.; Verma, V.; and Tang, J. 2019. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000*.

Suresh, S.; Li, P.; Hao, C.; and Neville, J. 2021. Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems*, 34: 15920–15933.

Tan, S.; Li, D.; Jiang, R.; Zhang, Y.; and Okumura, M. 2024. Community-invariant graph contrastive learning. *arXiv preprint arXiv:2405.01350*.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2018. Deep graph infomax. *arXiv preprint arXiv:1809.10341*.

Wang, S.; Wang, C.; Meng, P.; and Wang, Z. 2024a. AFANS: Augmentation-Free Graph Contrastive Learning with Adversarial Negative Sampling. In *International Conference on Intelligent Computing*, 376–387. Springer.

Wang, Y.; Min, Y.; Chen, X.; and Wu, J. 2021. Multi-view graph contrastive representation learning for drug-drug interaction prediction. In *Proceedings of the web conference 2021*, 2921–2933.

Wang, Z.; Hu, H.; He, C.; and Li, P. 2023a. Recognizing wafer map patterns using semi-supervised contrastive learning with optimized latent representation learning and data augmentation. In *2023 IEEE International Test Conference (ITC)*, 141–150. IEEE.

Wang, Z.; Liu, L.; Weston, S. R. F.; Tian, S.; and Li, P. 2024b. On learning discriminative features from synthesized data for self-supervised fine-grained visual recognition. In *European Conference on Computer Vision*, 101–117. Springer.

Wang, Z.; Somayaji, K. N.; and Li, P. 2024. Learn-by-Compare: Analog Performance Prediction using Contrastive Regression with Design Knowledge. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, 1–6.

Wang, Z.; Wang, Y.; Chen, Z.; Hu, H.; and Li, P. 2023b. Contrastive learning with consistent representations. *arXiv preprint arXiv:2302.01541*.

Wu, Y.; Zang, Z.; Zou, X.; Luo, W.; Bai, N.; Xiang, Y.; Li, W.; and Dong, W. 2025. Graph attention and Kolmogorov–Arnold network based smart grids intrusion detection. *Scientific Reports*, 15(1): 8648.

Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; and Pande, V. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical science*, 9(2): 513–530.

Xia, J.; Wu, L.; Wang, G.; Chen, J.; and Li, S. Z. 2021. Progcl: Rethinking hard negative mining in graph contrastive learning. *arXiv preprint arXiv:2110.02027*.

Xu, J.; Chen, Z.; Li, J.; Yang, S.; Wang, W.; Hu, X.; and Ngai, E. C. H. 2024. FourierKAN-GCF: Fourier Kolmogorov-Arnold Network – An Effective and Efficient Feature Transformation for Graph Collaborative Filtering. arXiv:2406.01034.

You, Y.; Chen, T.; Shen, Y.; and Wang, Z. 2021. Graph contrastive learning automated. In *International conference on machine learning*, 12121–12132. PMLR.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823.

Zhang, B.; Fan, C.; Liu, S.; Huang, K.; Zhao, X.; Huang, J.; and Liu, Z. 2024. The expressive power of graph neural networks: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Zhang, F.; and Zhang, X. 2024. GraphKAN: Enhancing Feature Extraction with Graph Kolmogorov Arnold Networks. arXiv:2406.13597.

Zhang, Q.; Yang, C.; and Shi, C. 2024. Adaptive negative representations for graph contrastive learning. *AI Open*, 5: 79–86.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021. Graph contrastive learning with adaptive augmentation. In *Proceedings of the web conference 2021*, 2069–2080.