Self-Supervised Contrastive Re-Learning for Multi-Graph Multi-Label Classification

Meixia Wang[1,2], Yuhai Zhao[1,2*], Zhengkui Wang[3], Yejiang Wang[1], Miaomiao Huang[1,2], Fenglong Ma[4], Fazal Wahab[1], Wen Shan[5], Xingwei Wang[1]

[1]School of Computer Science and Engineering, Northeastern University, China
[2]Key Laboratory of Intelligent Computing in Medical Image of Ministry of Education, Northeastern University, China
[3]InfoComm Technology Cluster, Singapore Institute of Technology, Singapore
[4]College of Information Sciences and Technology, Pennsylvania State University, USA
[5]Singapore University of Social Sciences, Singapore
wangmx7@mails.neu.edu.cn, {zhaoyuhai,wangxw}@mail.neu.edu.cn,
zhengkui.wang@singaporetech.edu.sg, fenglong@psu.edu, viviensw@suss.edu.sg

Abstract

Multi-graph multi-label learning (MGML) represents each object as a bag-of-graphs with multiple labels, but demands large-scale labeled data whose acquisition is often difficult and costly. Self-supervised contrastive learning (SCL) mitigates label dependence by leveraging data augmentation to construct discriminative pretext tasks, proving effective for multi-instance learning. However, when applied to MGML, SCL faces two key challenges: (1) it distinguishes individual instances by their differences, whereas MGML requires modeling label correlations; (2) it assumes semantic invariance under augmentation, but structural perturbations in MGML alter label semantics. To tackle these challenges, we propose a self-suPervised contrastive rE-learning framework for mulTi-grAph multi-labeL classification (PETAL). Specifically, to model label correlations, we first define a unified label space to learn label prototypes and align features with them, yielding prototype-aligned representations. We then design a multi-granularity contrastive loss over these representations, which captures label dependencies by contrasting at the bag level, graph level, and bag-graph level. Moreover, to ensure semantic invariance, we develop a contrastive re-learning strategy based on prototype-aligned representations to generate augmentation-free positive samples. This guarantees consistent multi-label distributions without structural perturbations. Experiments on six datasets demonstrate that PETAL achieves an average improvement of 4.12% over state-of-the-art self-supervised and supervised baselines.

Introduction

Supervised learning, a fundamental machine learning paradigm, typically represents objects as single instances (Zhao et al. 2025a) or as bags-of-instances in multi-instance learning (MIL) (Zhang et al. 2024). While MIL advances beyond single-instance by capturing fine-grained details, its bag representation inherently neglects structural information, hindering the modeling of complex objects with rich topology (Bian et al. 2024; Zhang et al. 2025). In contrast, graphs explicitly encode structural relationships, providing superior modeling capabilities (Wu et al. 2025; Zhang, Yuan, and Pan 2024). Existing graph learning methods fall into two types: single-graph learning (Zhuo et al. 2025b; Yang et al. 2025; Zhuo et al. 2025a) captures global structures but ignores fine-grained details, while multi-graph learning (MGL) (Wu et al. 2017) represents objects as bags-of-graphs to capture both fine-grained details and global relationships. Despite its advantages, early MGL methods primarily focus on single-label scenarios, whereas multi-label learning is crucial for many applications. Multi-graph multi-label learning (MGML) (Zhu and Zhao 2018) addresses this by associating bags-of-graphs with multiple labels, enabling more complex applications such as computing power networks.

Existing MGML methods show strong performance but rely heavily on multi-label annotated datasets (Wang et al. 2024). Due to the high cost of annotation, collecting fully annotated MGML data is difficult. To mitigate this, self-supervised contrastive learning (SCL) learns representations from unlabeled data via pretext tasks (Xue et al. 2025), reducing reliance on labeled data. SCL minimizes the distance between augmented views of the same sample while maximizing it across different samples. Although simple and effective, SCL mainly suits single-label tasks such as single-instance, multi-instance, and single-graph single-label learning (Sun et al. 2024; Feng et al. 2025). Its adaptation to MGML scenarios remains largely unexplored and faces two key challenges: (1) modeling complex label correlations and (2) ensuring semantic invariance under augmentations.

The first challenge lies in effectively modeling label correlations. SCL has been widely applied to single-label tasks, achieving strong performance across various domains (Peng et al. 2024), which naturally motivates its extension to multi-label settings such as MGML. However, this adaptation is far from trivial (Lin et al. 2023). In single-label scenarios, each sample has a unique, mutually exclusive label, allowing SCL to effectively distinguish individual instances based on their differences. In contrast, in multi-label scenarios like MGML, each sample is associated with multiple correlated labels, necessitating effective modeling of label correlations.

*Corresponding author.

Nevertheless, conventional SCL focuses on distinguishing instances by pulling positive pairs together and pushing negative pairs apart, which limits its ability to capture label correlations. Therefore, adapting SCL to MGML requires incorporating components that model label dependencies.

The second challenge lies in ensuring semantic invariance. SCL typically constructs positive pairs through data augmentation. This strategy relies on the assumption that augmented views preserve semantic consistency with the original samples. While this assumption is often valid in image or text domains, it is frequently violated for graph data, primarily due to the discrete nature of graph structures, where even minor perturbations can lead to significant semantic changes (Wang et al. 2023b; Xiao et al. 2023). Consequently, semantic invariance between original and augmented graphs may be compromised (Zhao et al. 2025b). This challenge is also prevalent in MGML, where multi-graph augmentation is implemented through structural or numerical perturbations, which may compromise semantic invariance at both the graph and multi-graph bag levels. Therefore, to adapt SCL for MGML, it is essential to ensure the generation of semantic invariant positive pairs.

To address the above challenges, we propose a self-suPervised contrastive rE-learning framework for mulTi-grAph multi-labeL classification (PETAL). Specifically, to model label correlations, we first extract dual-granularity feature representations at both the bag and graph levels using graph kernels and an encoder-decoder architecture. We then define a unified label space and introduce a prototype encoder to learn a prototype for each label, which serves as a class centroid to compute distances between sample features and label prototypes, resulting in prototype-aligned representations. Based on these representations, we further design a multi-granularity contrastive loss by contrasting at the bag level, graph level, and bag-graph level, enabling the modeling of complex label correlations at different granularities. Furthermore, to ensure semantic invariance, we develop a contrastive re-learning strategy that regenerates positive samples without relying on augmentations by using prototype-aligned representations. This approach guarantees consistent multi-label distributions between anchors and positives, while avoiding structural perturbations that could alter semantic labels. The key contributions of this work are summarized as follows:

- We propose a self-supervised learning paradigm, PETAL, for MGML, which is more practical and effective than existing supervised learning methods. To the best of our knowledge, this is the first attempt to learn multi-graph bag representations in a self-supervised manner for downstream multi-label classification tasks.
- Our proposed PETAL method introduces a multi-granularity contrastive loss to model label correlations and a contrastive re-learning strategy to ensure semantic invariance between positive pairs, thereby effectively adapting SCL to MGML scenarios.
- Extensive experiments demonstrate that the proposed PETAL algorithm significantly outperforms state-of-the-art self-supervised and supervised learning baselines.

Related Work

Multi-Graph Multi-Label Learning. MGML, first proposed by (Zhu and Zhao 2018) as an extension of MGL (Wu et al. 2014) to multi-label scenarios, represents graph bags as binary vectors derived from informative subgraph mining. However, this binarization leads to the loss of structural information. To address this, recent studies use graph kernel methods to better preserve structure (Zhao et al. 2021; Wang et al. 2023a). This approach has been extended to scenarios with novel, missing (Huang et al. 2023), or noisy labels (Wang et al. 2024). Despite advances, existing methods still rely on large labeled datasets, which are costly to obtain.

Self-Supervised Contrastive Learning. SCL, a branch of self-supervised learning, reduces manual annotation needs while maintaining competitive performance. Recent advances have further narrowed its gap with supervised methods. Representative approaches such as GraphCL (You et al. 2020) and JOAO (You et al. 2021) pull augmented views of the same sample closer and push apart those of different samples (Xia et al. 2022). Methods like BYOL (Grill et al. 2020) and SimSiam (Chen and He 2021) prevent collapse using asymmetric architectures and moving averages. Other variants exploit clustering (Asano, Rupprecht, and Vedaldi 2020), optimal transport (Wang et al. 2023b), or temporal consistency (Zhao et al. 2025b). However, applying SCL to MGML remains unexplored, with two key challenges: modeling label correlations and ensuring semantic invariance.

Methodology

We first overview PETAL (Fig. 1), which learns bag representations via a pretext task without ground-truth labels for downstream multi-label classification. PETAL consists of three core modules: dual-granularity feature extraction, label-specific prototype representation, and multi-granularity contrastive re-learning. Specifically, the dual-granularity feature extraction module uses graph kernels (I) and a multi-graph encoder-decoder to obtain fine-grained graph representations (II), aggregated into coarse-grained bag representations (III). The label-specific prototype representation module defines a unified label space and uses a prototype encoder to learn label prototypes (IV). By measuring distances between sample features and these prototypes, it generates prototype-aligned representations at both the graph and bag levels (V, VI). The multi-granularity contrastive re-learning module comprises two key components: a multi-granularity contrastive loss and a contrastive re-learning strategy (f_c). The former captures label correlations, while the latter generates new views of prototype-aligned representations as positive samples, ensuring consistent multi-label distributions and semantic invariance.

Problem Definition

In MGML, a dataset consists of n bags $\mathcal{B} = \{B_1, B_2, ..., B_n\}$, and their corresponding label vectors $\mathcal{Y} = \{\boldsymbol{Y}_1, \boldsymbol{Y}_2, ..., \boldsymbol{Y}_n\}$, where each $\boldsymbol{Y}_i \in \{0, 1\}^L$ denotes the labels assigned over L classes. Each bag $B_i = \{G_{i1}, G_{i2}, ..., G_{in_i}\}$ represents a sample containing n_i graphs. The label vector $\boldsymbol{Y}_i = [y_{i1}, y_{i2}, ..., y_{iL}]^{\mathrm{T}}$ indicates bag B_i has label j if $y_{ij} = 1$, and

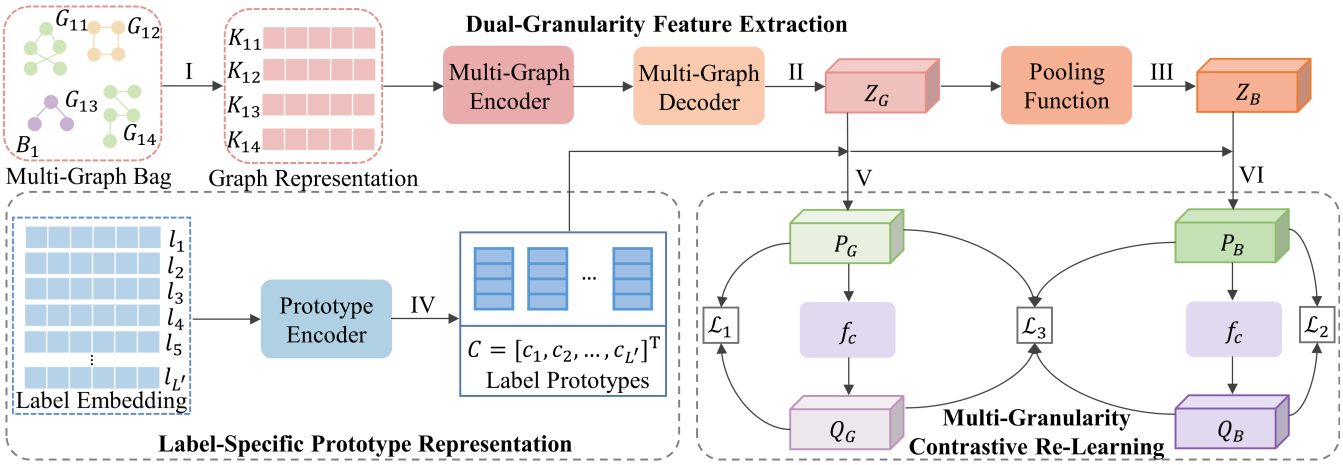

Figure 1: Framework of the proposed PETAL method.

lacks it otherwise. Each graph $G_{ij} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ is defined by its node set $\mathcal{V}$, edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and node features $\mathcal{X}$. The objective of MGML is to learn a mapping $f : \mathcal{B} \to \mathcal{Y}$ that predicts labels for unseen bags.

Dual-Granularity Feature Extraction

Different from traditional single-instance learning, single-graph learning, MIL, and multi-instance multi-label learning, MGML represents each object as a bag-of-graphs that inherently contains both coarse-grained (bag-level) and fine-grained (graph-level) information. To fully leverage this structure, it is essential to capture representations at both the bag-level and graph-level when applying SCL to MGML. However, existing MGML methods often convert graphs into binary vectors by extracting frequent or informative subgraphs, failing to preserve intrinsic structural information. To mitigate this issue, we follow (Zhao et al. 2021) by embedding graphs into a Hilbert space using graph kernels. Similarities with anchor graphs yield kernel vectors, serving as initial representations to preserve intra-graph and inter-graph structures. Nevertheless, these representations are statically defined by kernel functions and are non-learnable (Sun and Fan 2024; Sun and Kok 2025). To overcome this, we design a trainable multi-graph encoder-decoder architecture that produces learnable graph representations, jointly optimized and aggregated into bag-level representations.

Formally, the graph kernel is a symmetric positive-definite function $k : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ that corresponds to an inner product in a reproducing kernel Hilbert space $\mathcal{H}$. For any graphs $G_1, G_2 \in \mathcal{G}$, there exists a feature map $\phi : \mathcal{G} \to \mathcal{H}$ satisfying $k(G_1, G_2) = \langle \phi(G_1), \phi(G_2) \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the inner product. This equivalence enables the kernel trick: high-dimensional inner products are computed implicitly via k in the original space. In MGML, the kernel value for graphs G_i, G_j is given by Eq. (1).

$$k(G_i, G_j) = \langle \phi(G_i), \phi(G_j) \rangle \tag{1}$$

Thus, each graph G_{ij} in bag B_i is represented by a similarity kernel vector $\boldsymbol{K}_{ij} \in \mathbb{R}^t$, as expressed in Eq. (2).

$$\boldsymbol{K}_{ij} = [k(G_{ij}, G_1), k(G_{ij}, G_2), ..., k(G_{ij}, G_t)]^{\mathrm{T}} \tag{2}$$

In Eq. (2), t denotes the number of template graphs sampled from the training set, e.g., a subset or all training graphs can be used as templates. This kernel-based initial graph representation preserves both intra-graph structural information and inter-graph similarities. Consequently, we obtain a transformed MGML dataset $\mathcal{B}' = \{\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_n\}$ with $\mathcal{K}_i = \{\boldsymbol{K}_{i1}, \boldsymbol{K}_{i2}, \ldots, \boldsymbol{K}_{in_i}\}$.

However, traditional graph kernel functions are static and non-trainable, causing the kernel-based initial graph representation $\boldsymbol{K}_{ij}$ to be entirely dependent on the chosen kernel k. To address this limitation, we employ a multi-graph encoder-decoder architecture to learn trainable graph representations $\boldsymbol{Z}_{G_{ij}}$, as formalized in Eq. (3).

$$\boldsymbol{Z}_{G_{ij}} = f_\gamma(f_\theta(\boldsymbol{K}_{ij})) \tag{3}$$

In Eq. (3), f_θ and f_γ are MLP-based encoder and decoder; the decoder reconstructs topology-faithful representations and projects them into a prototype-aligned space.

It is worth noting that MGML represents each object as a bag-of-graphs, and bag feature representations are derived from the constituent graphs within these bags. Following the standard MGML assumption, if a bag contains at least one graph with label i, it inherits label i. We model graph-bag relationships using max pooling. The representation of bag B_i is given by Eq. (4).

$$\boldsymbol{Z}_{B_i} = \max_{j \in n_i} (\boldsymbol{Z}_{G_{ij}}) \tag{4}$$

In Eq. (4), $G_{ij} \in B_i$. This operation retains the most informative graph features in each dimension, enabling the dual-granularity extraction module to produce trainable graph and bag representations while preserving structural information.

Label-Specific Prototype Representation

In single-label scenarios, each sample is exclusively associated with one label, necessitating the learning of inter-label distinctions (e.g., an image labeled "cat" cannot be "dog"). In contrast, multi-label scenarios allow multiple interdependent labels per sample (e.g., "polar bear" often co-occurs with "glacier" but rarely with "penguin"), requiring not only

discriminative representation learning but also explicit modeling of label correlations. These challenges significantly complicate the adaptation of SCL to MGML tasks.

Therefore, effectively applying SCL to MGML tasks requires bag representations that capture both discriminative features and latent label correlations. However, most existing SCL methods are designed for single-label scenarios, where they distinguish instances by pulling together augmented views of the same sample while pushing apart different samples. Such methods inherently fail to model the label dependencies critical for MGML performance. Although supervised learning methods can capture label correlations, they depend on ground-truth annotations during training, which are unavailable in self-supervised settings. It is thus imperative to design pretext tasks that enable learning of label correlations without relying on annotations.

Inspired by unsupervised clustering and SwAV (Caron et al. 2018), we learn label prototypes as class cluster centers to implicitly capture correlations through feature-prototype distance measurements. Given the unavailability of ground-truth labels, we define a unified label space with L' pseudo-labels, initializing each label embedding l_i as a one-hot vector with 1 at position i and 0 elsewhere, as shown in Eq. (5).

$$\boldsymbol{l}_i = [0, 0, ..., 1, ..., 0]^{\mathrm{T}} \tag{5}$$

To learn label-specific prototype representations, we employ an MLP-based prototype encoder f_δ to encode the initialized label embeddings, as shown in Eq. (6).

$$\boldsymbol{c}_i = f_\delta(\boldsymbol{l}_i) \tag{6}$$

Here, $\boldsymbol{c}_i$ denotes the prototype of label i. The label prototype matrix is defined as $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{L'}]^\top \in \mathbb{R}^{L' \times d_L}$, where d_L is the dimension of each label prototype.

We then leverage the distances between label prototypes and feature representations at both the bag and graph levels to construct prototype-aligned dual-granularity representations, as defined in Eqs. (7) and (8).

$$\boldsymbol{P}_G = softmax(\boldsymbol{Z}_G \cdot \boldsymbol{C}/\tau) \tag{7}$$

$$\boldsymbol{P}_B = softmax(\boldsymbol{Z}_B \cdot \boldsymbol{C}/\tau) \tag{8}$$

Here, $\boldsymbol{Z} \cdot \boldsymbol{C}$ represents the distance between the feature representation and the label-specific prototype representation, while τ denotes the temperature parameter.

Multi-Granularity Contrastive Re-Learning

In SCL, it is fundamental to define positive and negative samples. While negative samples are typically drawn from other samples, positive samples are constructed under the assumption of semantic consistency with the anchor. However, this assumption often fails for graph data due to its discrete structure and the sensitivity of semantics to structural changes. Even minor perturbations can disrupt critical substructures, altering labels and making it difficult to generate semantically consistent positive samples. This issue is also prevalent in MGML, where bag labels are defined by the union of constituent graph labels. The removal of a graph or modification of key structures can directly alter the bag's label distribution. Therefore, generating positive samples that preserve label semantics without relying on augmentation is essential for effective SCL in MGML.

To address these challenges, we propose a contrastive re-learning strategy that generates positive samples with consistent multi-label distributions through re-learning prototype-aligned anchor representations, rather than directly perturbing the input data. This approach circumvents the uncontrolled semantic distortions caused by perturbing complex graph structures, ensuring that the positives share the same semantic entity as the anchors and eliminating semantic inconsistencies and distributional shifts. In this work, we instantiate this strategy using an optimal transport-based re-learning algorithm f_c (Cuturi 2013), which computes the minimal-cost transport plan between anchor representations and learned semantic prototypes $\boldsymbol{C}$. This approach preserves distributional consistency and leverages the stable semantic information captured by the prototypes to generate semantically aligned positive samples. Notably, while we employ optimal transport in this work, contrastive re-learning is a general strategy applicable across modalities and tasks with diverse generation mechanisms. This strategy is specifically implemented as formulated in Eqs. (9) and (10).

$$\boldsymbol{Q}_G = f_c(\boldsymbol{Z}_G, \boldsymbol{C}) \tag{9}$$

$$\boldsymbol{Q}_B = f_c(\boldsymbol{Z}_B, \boldsymbol{C}) \tag{10}$$

In the above equations, $\boldsymbol{Q}_G$ and $\boldsymbol{Q}_B$ denote the generated positive samples at the graph and bag levels, respectively.

We then compute the multi-granularity contrastive loss over augmentation-free positive pairs to capture label correlations while preserving semantic invariance. The loss comprises three components: coarse-grained bag-level, fine-grained graph-level, and cross-granularity contrastive losses. For both bag-level (Eq. 12) and graph-level (Eq. 11) terms, the prototype-aligned representations at each granularity serve as anchors. Their positive samples are generated by the re-learning function f_c, while the re-learned counterparts of other samples serve as negatives. By pulling positives closer and pushing negatives apart, these losses jointly enhance feature discrimination and model label dependencies.

$$\mathcal{L}_1 = CE(\boldsymbol{P}_G, \boldsymbol{Q}_G) \tag{11}$$

$$\mathcal{L}_2 = CE(\boldsymbol{P}_B, \boldsymbol{Q}_B) \tag{12}$$

In the above equations, $CE(\cdot)$ is the cross-entropy loss.

Furthermore, we introduce cross-granularity contrastive loss between graph-level and bag-level representations to integrate complementary semantic cues across granularities, as shown in Eq. (13).

$$\mathcal{L}_3 = CE(\rho(\boldsymbol{P}_G), \boldsymbol{Q}_B) \tag{13}$$

In Eq. (13), ρ denotes the max pooling function projecting graph representations to the bag representation space, where the bag representation serves as the anchor. Its positive samples are the re-learned graph representations within the same bag, while the re-learned graphs from other bags act as negatives. The final multi-granularity contrastive loss aggregates the components as shown in Eq. (14).

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 \tag{14}$$

Overall, PETAL has time complexity $O(NL)$ and space complexity $O(Nt+NL)$, where N is the number of graphs, L the number of labels, and t the number of template graphs.

Datasets	M3MIML	KISAR	MIMLFast	MIML-LLMC	cfMGML	cfMGNML	PETAL
One Error ↓							
MSRC v2	0.552 ± 0.050	0.514 ± 0.070	0.383 ± 0.037	0.309 ± 0.089	0.334 ± 0.070	0.323 ± 0.047	**0.308 ± 0.054**
Scene	0.559 ± 0.032	0.337 ± 0.032	0.432 ± 0.024	0.796 ± 0.024	0.309 ± 0.102	0.329 ± 0.100	**0.299 ± 0.032**
VOC12	0.280 ± 0.059	**0.278 ± 0.056**	0.410 ± 0.061	0.288 ± 0.054	0.284 ± 0.039	0.310 ± 0.059	**0.278 ± 0.038**
Dblp(cv)	0.317 ± 0.044	0.421 ± 0.053	0.329 ± 0.037	0.272 ± 0.025	0.291 ± 0.056	**0.270 ± 0.068**	**0.270 ± 0.033**
Dblp(ai)	0.439 ± 0.034	0.510 ± 0.075	0.365 ± 0.044	0.320 ± 0.059	0.369 ± 0.053	0.317 ± 0.037	**0.272 ± 0.036**
Dblp(db)	0.320 ± 0.025	0.323 ± 0.033	0.382 ± 0.038	0.342 ± 0.079	0.396 ± 0.126	0.344 ± 0.074	**0.294 ± 0.042**
Hamming Loss ↓							
MSRC v2	0.118 ± 0.009	0.082 ± 0.011	0.187 ± 0.025	0.081 ± 0.009	0.179 ± 0.019	0.551 ± 0.016	**0.078 ± 0.008**
Scene	0.271 ± 0.012	0.180 ± 0.010	0.270 ± 0.024	0.247 ± 0.006	0.337 ± 0.065	0.301 ± 0.039	**0.166 ± 0.016**
VOC12	**0.093 ± 0.005**	0.094 ± 0.006	0.186 ± 0.020	0.094 ± 0.006	0.247 ± 0.021	0.567 ± 0.010	**0.093 ± 0.006**
Dblp(cv)	0.101 ± 0.006	0.091 ± 0.004	0.177 ± 0.022	0.088 ± 0.007	0.255 ± 0.035	0.648 ± 0.027	**0.087 ± 0.006**
Dblp(ai)	0.120 ± 0.006	0.099 ± 0.010	0.158 ± 0.012	0.098 ± 0.007	0.258 ± 0.017	0.690 ± 0.022	**0.090 ± 0.006**
Dblp(db)	0.119 ± 0.010	0.110 ± 0.009	0.232 ± 0.020	0.111 ± 0.013	0.270 ± 0.029	0.752 ± 0.017	**0.104 ± 0.008**
Coverage ↓							
MSRC v2	0.488 ± 0.028	0.299 ± 0.038	0.311 ± 0.024	0.530 ± 0.032	0.317 ± 0.031	0.274 ± 0.016	**0.240 ± 0.023**
Scene	0.339 ± 0.027	0.197 ± 0.015	0.239 ± 0.020	0.426 ± 0.019	0.189 ± 0.073	0.172 ± 0.049	**0.170 ± 0.014**
VOC12	0.451 ± 0.026	0.546 ± 0.023	0.440 ± 0.021	0.511 ± 0.040	0.423 ± 0.032	0.379 ± 0.037	**0.345 ± 0.016**
Dblp(cv)	0.333 ± 0.026	0.323 ± 0.041	0.252 ± 0.016	0.332 ± 0.081	0.205 ± 0.028	0.181 ± 0.023	**0.177 ± 0.017**
Dblp(ai)	0.259 ± 0.021	0.335 ± 0.026	0.223 ± 0.025	0.312 ± 0.048	0.209 ± 0.015	0.182 ± 0.026	**0.163 ± 0.024**
Dblp(db)	0.287 ± 0.028	0.240 ± 0.016	0.280 ± 0.020	0.426 ± 0.042	0.295 ± 0.035	0.248 ± 0.014	**0.215 ± 0.022**
Ranking Loss ↓							
MSRC v2	0.305 ± 0.024	0.163 ± 0.021	0.155 ± 0.013	0.387 ± 0.065	0.148 ± 0.014	0.131 ± 0.020	**0.110 ± 0.011**
Scene	0.357 ± 0.029	0.178 ± 0.014	0.229 ± 0.019	-	0.162 ± 0.057	0.148 ± 0.034	**0.147 ± 0.015**
VOC12	0.234 ± 0.014	0.355 ± 0.023	0.253 ± 0.019	0.334 ± 0.051	0.221 ± 0.022	0.196 ± 0.021	**0.176 ± 0.011**
Dblp(cv)	0.196 ± 0.019	0.197 ± 0.034	0.135 ± 0.012	0.264 ± 0.086	0.111 ± 0.023	**0.093 ± 0.018**	**0.093 ± 0.017**
Dblp(ai)	0.176 ± 0.013	0.260 ± 0.027	0.137 ± 0.016	0.233 ± 0.055	0.124 ± 0.011	0.102 ± 0.016	**0.089 ± 0.017**
Dblp(db)	0.179 ± 0.022	0.143 ± 0.016	0.179 ± 0.019	0.342 ± 0.079	0.178 ± 0.041	0.142 ± 0.024	**0.120 ± 0.020**
Average Precision ↑							
MSRC v2	0.453 ± 0.020	0.557 ± 0.033	0.603 ± 0.020	0.553 ± 0.057	0.631 ± 0.040	0.659 ± 0.030	**0.696 ± 0.035**
Scene	0.619 ± 0.021	0.782 ± 0.020	0.719 ± 0.017	0.512 ± 0.017	0.795 ± 0.066	0.796 ± 0.051	**0.814 ± 0.018**
VOC12	0.567 ± 0.018	0.483 ± 0.033	0.519 ± 0.028	0.557 ± 0.030	0.589 ± 0.026	0.589 ± 0.039	**0.623 ± 0.018**
Dblp(cv)	0.686 ± 0.026	0.603 ± 0.026	0.709 ± 0.018	0.699 ± 0.043	0.744 ± 0.041	0.762 ± 0.041	**0.769 ± 0.027**
Dblp(ai)	0.642 ± 0.019	0.522 ± 0.046	0.701 ± 0.022	0.690 ± 0.049	0.715 ± 0.022	0.749 ± 0.020	**0.780 ± 0.027**
Dblp(db)	0.706 ± 0.021	0.720 ± 0.024	0.673 ± 0.026	0.635 ± 0.052	0.668 ± 0.077	0.709 ± 0.045	**0.747 ± 0.032**
Macro-F1 ↑							
MSRC v2	0.020 ± 0.007	0.278 ± 0.040	0.320 ± 0.024	0.194 ± 0.042	0.327 ± 0.031	0.243 ± 0.016	**0.419 ± 0.049**
Scene	0.011 ± 0.009	0.535 ± 0.025	0.533 ± 0.024	-	0.279 ± 0.054	0.336 ± 0.094	**0.641 ± 0.035**
VOC12	0.042 ± 0.002	0.066 ± 0.018	0.171 ± 0.019	0.077 ± 0.020	0.198 ± 0.026	0.194 ± 0.020	**0.199 ± 0.045**
Dblp(cv)	0.076 ± 0.021	0.208 ± 0.021	**0.309 ± 0.019**	0.202 ± 0.045	0.307 ± 0.047	0.230 ± 0.024	0.244 ± 0.027
Dblp(ai)	0.055 ± 0.020	0.256 ± 0.042	0.382 ± 0.027	0.308 ± 0.034	0.382 ± 0.038	0.242 ± 0.013	**0.386 ± 0.033**
Dblp(db)	0.067 ± 0.018	0.318 ± 0.053	0.384 ± 0.021	0.306 ± 0.061	0.370 ± 0.039	0.260 ± 0.014	**0.397 ± 0.051**

Table 1: Comparison with supervised learning baselines on the multi-graph multi-label classification task. Bold indicates optimal results, underlined indicates sub-optimal results. ↑: higher is better, ↓: lower is better. "-": not applicable.

Experiment

We evaluated the proposed PETAL method on six benchmark datasets for MGML, namely MSRC v2, VOC12, Scene, Dblp(ai), Dblp(cv), and Dblp(db) (Zhao et al. 2021). The performance was assessed using six standard multi-label evaluation metrics, namely One Error, Hamming Loss, Coverage, Ranking Loss, Average Precision, and Macro-averaging F1 (Macro-F1) (Zhang and Zhou 2014). To verify the effectiveness of PETAL, we compared it against twelve representative baseline methods, covering both supervised and self-supervised learning paradigms. The supervised baselines include four MIML algorithms, namely M3MIML (Zhang and Zhou 2008), KISAR (Li et al. 2012), MIMLFast (Huang, Gao, and Zhou 2019), and MIML-LLMC (Yang, Tang, and Min 2022), as well as two MGML methods, cfMGML (Zhao et al. 2021) and cfMGNML (Wang et al. 2024). The self-supervised baselines comprise five self-supervised graph contrastive learning methods, in-

Datasets	GRAPHCL	JOAO v2	SimGRACE	GALOPA	PaGCL	SMILES	PETAL
One Error ↓							
MSRC v2	0.514 ± 0.042	0.485 ± 0.032	0.491 ± 0.024	0.484 ± 0.030	0.433 ± 0.020	0.667 ± 0.023	**0.404 ± 0.021**
Scene	0.600 ± 0.007	0.473 ± 0.013	0.531 ± 0.014	0.456 ± 0.008	0.468 ± 0.012	0.697 ± 0.018	**0.346 ± 0.011**
VOC12	0.281 ± 0.006	0.280 ± 0.012	0.281 ± 0.007	0.281 ± 0.011	0.285 ± 0.004	0.321 ± 0.015	**0.278 ± 0.007**
Dblp(cv)	0.545 ± 0.008	0.545 ± 0.009	0.544 ± 0.017	0.541 ± 0.017	0.528 ± 0.003	0.446 ± 0.005	**0.287 ± 0.009**
Dblp(ai)	0.640 ± 0.023	0.639 ± 0.014	0.634 ± 0.009	0.636 ± 0.005	0.656 ± 0.029	0.496 ± 0.019	**0.326 ± 0.018**
Dblp(db)	0.633 ± 0.009	0.633 ± 0.008	0.645 ± 0.022	0.650 ± 0.009	0.655 ± 0.017	0.483 ± 0.018	**0.338 ± 0.016**
Hamming Loss ↓							
MSRC v2	0.106 ± 0.002	0.103 ± 0.002	0.104 ± 0.003	0.102 ± 0.002	0.098 ± 0.001	0.148 ± 0.005	**0.091 ± 0.002**
Scene	0.244 ± 0.002	0.224 ± 0.002	0.238 ± 0.002	0.225 ± 0.003	0.223 ± 0.003	0.340 ± 0.008	**0.185 ± 0.004**
VOC12	**0.093 ± 0.001**	**0.093 ± 0.001**	0.094 ± 0.001	**0.093 ± 0.001**	0.094 ± 0.001	0.101 ± 0.002	0.094 ± 0.001
Dblp(cv)	0.125 ± 0.001	0.124 ± 0.002	0.123 ± 0.002	0.124 ± 0.001	0.123 ± 0.002	0.118 ± 0.002	**0.091 ± 0.002**
Dblp(ai)	0.132 ± 0.001	0.131 ± 0.001	0.131 ± 0.002	0.132 ± 0.002	0.134 ± 0.001	0.121 ± 0.001	**0.096 ± 0.001**
Dblp(db)	0.148 ± 0.001	0.148 ± 0.001	0.148 ± 0.001	0.149 ± 0.002	0.149 ± 0.003	0.134 ± 0.004	**0.113 ± 0.003**
Coverage ↓							
MSRC v2	0.387 ± 0.010	0.378 ± 0.013	0.376 ± 0.006	0.348 ± 0.021	0.312 ± 0.014	0.557 ± 0.015	**0.309 ± 0.019**
Scene	0.338 ± 0.002	0.260 ± 0.005	0.288 ± 0.006	0.255 ± 0.006	0.261 ± 0.008	0.399 ± 0.006	**0.195 ± 0.005**
VOC12	0.413 ± 0.004	0.409 ± 0.005	0.424 ± 0.006	0.404 ± 0.008	0.428 ± 0.011	0.466 ± 0.006	**0.380 ± 0.009**
Dblp(cv)	0.287 ± 0.005	0.288 ± 0.007	0.288 ± 0.004	0.290 ± 0.003	0.289 ± 0.004	0.293 ± 0.005	**0.202 ± 0.003**
Dblp(ai)	0.361 ± 0.004	0.362 ± 0.008	0.369 ± 0.009	0.370 ± 0.012	0.377 ± 0.004	0.273 ± 0.005	**0.188 ± 0.007**
Dblp(db)	0.440 ± 0.009	0.444 ± 0.011	0.444 ± 0.011	0.452 ± 0.015	0.448 ± 0.005	0.343 ± 0.011	**0.239 ± 0.006**
Ranking Loss ↓							
MSRC v2	0.215 ± 0.009	0.212 ± 0.013	0.210 ± 0.007	0.195 ± 0.013	0.162 ± 0.007	0.360 ± 0.009	**0.160 ± 0.013**
Scene	0.361 ± 0.002	0.261 ± 0.007	0.297 ± 0.006	0.254 ± 0.005	0.263 ± 0.010	0.442 ± 0.009	**0.178 ± 0.006**
VOC12	0.218 ± 0.003	0.214 ± 0.004	0.224 ± 0.007	0.212 ± 0.004	0.229 ± 0.005	0.257 ± 0.004	**0.198 ± 0.006**
Dblp(cv)	0.190 ± 0.005	0.190 ± 0.006	0.190 ± 0.003	0.193 ± 0.004	0.191 ± 0.003	0.191 ± 0.003	**0.111 ± 0.004**
Dblp(ai)	0.283 ± 0.007	0.284 ± 0.006	0.289 ± 0.008	0.292 ± 0.011	0.299 ± 0.004	0.193 ± 0.007	**0.110 ± 0.006**
Dblp(db)	0.331 ± 0.009	0.334 ± 0.011	0.336 ± 0.012	0.346 ± 0.013	0.341 ± 0.006	0.240 ± 0.011	**0.141 ± 0.006**
Average Precision ↑							
MSRC v2	0.515 ± 0.014	0.532 ± 0.016	0.529 ± 0.009	0.546 ± 0.012	0.594 ± 0.016	0.372 ± 0.006	**0.607 ± 0.006**
Scene	0.607 ± 0.004	0.699 ± 0.008	0.659 ± 0.007	0.708 ± 0.005	0.701 ± 0.008	0.536 ± 0.010	**0.781 ± 0.007**
VOC12	0.578 ± 0.003	0.583 ± 0.007	0.572 ± 0.010	0.584 ± 0.005	0.568 ± 0.005	0.535 ± 0.005	**0.602 ± 0.006**
Dblp(cv)	0.593 ± 0.004	0.593 ± 0.005	0.593 ± 0.009	0.593 ± 0.009	0.597 ± 0.003	0.621 ± 0.003	**0.746 ± 0.006**
Dblp(ai)	0.466 ± 0.005	0.467 ± 0.005	0.464 ± 0.007	0.465 ± 0.009	0.455 ± 0.010	0.607 ± 0.011	**0.739 ± 0.012**
Dblp(db)	0.485 ± 0.005	0.483 ± 0.003	0.474 ± 0.010	0.471 ± 0.008	0.470 ± 0.012	0.588 ± 0.015	**0.713 ± 0.011**
Macro-F1 ↑							
MSRC v2	0.261 ± 0.044	0.263 ± 0.012	0.257 ± 0.027	0.278 ± 0.035	**0.331 ± 0.009**	0.138 ± 0.011	0.330 ± 0.006
Scene	0.326 ± 0.009	0.436 ± 0.011	0.384 ± 0.018	0.450 ± 0.008	0.451 ± 0.010	0.298 ± 0.013	**0.591 ± 0.011**
VOC12	0.129 ± 0.010	0.134 ± 0.016	0.126 ± 0.007	0.133 ± 0.014	0.130 ± 0.007	0.122 ± 0.006	**0.168 ± 0.009**
Dblp(cv)	0.075 ± 0.009	0.077 ± 0.011	0.083 ± 0.011	0.085 ± 0.005	0.091 ± 0.005	0.171 ± 0.017	**0.223 ± 0.016**
Dblp(ai)	0.098 ± 0.013	0.109 ± 0.005	0.108 ± 0.017	0.115 ± 0.008	0.109 ± 0.009	0.270 ± 0.011	**0.356 ± 0.026**
Dblp(db)	0.114 ± 0.024	0.099 ± 0.017	0.101 ± 0.005	0.110 ± 0.026	0.117 ± 0.015	0.272 ± 0.014	**0.380 ± 0.013**

Table 2: Comparison with self-supervised learning baselines on the multi-graph multi-label classification task. Bold indicates optimal results, underlined indicates sub-optimal results. ↑: higher is better, ↓: lower is better.

cluding GRAPHCL (You et al. 2020), JOAO v2 (You et al. 2021), SimGRACE (Xia et al. 2022), GALOPA (Wang et al. 2023b), and PaGCL (Zhao et al. 2025b), as well as one self-supervised MIL method, namely SMILES (Wang et al. 2023c). In this paper, f_c employs optimal transport, a deterministic, gradient-free process that utilizes a dot-product cost matrix with $\epsilon = 0.05$, balanced by Sinkhorn, which prevents collapse. Experiments were conducted on an A5000 GPU with the following hyperparameter settings: batch size of 128, learning rate of 1×10^{-5}, $L^{'}$ set to 200, and τ set to 0.09. For kernel selection, the Weisfeiler-Lehman kernel (Shervashidze et al. 2011) was used on the MSRC v2 dataset, while the Graph-Hopper kernel (Feragen et al. 2013) was used on the other datasets.

Comparison with Supervised Baselines

We first evaluated the effectiveness of PETAL by comparing it with supervised baselines, including four advanced

Methods	One Error ↓	Hamming Loss ↓	Coverage ↓	Ranking Loss ↓	Average Precision ↑	Macro-F1 ↑
w/o $\mathcal{L}_1$	$\underline{0.277 \pm 0.043}$	$\mathbf{0.090 \pm 0.010}$	$\mathbf{0.163 \pm 0.028}$	$\mathbf{0.089 \pm 0.019}$	$\underline{0.778 \pm 0.036}$	$\underline{0.379 \pm 0.038}$
w/o $\mathcal{L}_2$	0.286 ± 0.034	$\underline{0.092 \pm 0.007}$	$\underline{0.169 \pm 0.022}$	$\underline{0.093 \pm 0.011}$	0.772 ± 0.019	0.372 ± 0.039
w/o $\mathcal{L}_3$	0.303 ± 0.066	0.094 ± 0.009	0.181 ± 0.029	0.104 ± 0.025	0.752 ± 0.045	0.349 ± 0.041
w/o f_c	0.691 ± 0.047	0.138 ± 0.004	0.408 ± 0.023	0.329 ± 0.022	0.430 ± 0.027	0.109 ± 0.027
w/o $\boldsymbol{C}$	0.346 ± 0.028	0.107 ± 0.005	0.203 ± 0.013	0.120 ± 0.012	0.721 ± 0.018	0.327 ± 0.035
PETAL	$\mathbf{0.272 \pm 0.036}$	$\mathbf{0.090 \pm 0.006}$	$\mathbf{0.163 \pm 0.024}$	$\mathbf{0.089 \pm 0.017}$	$\mathbf{0.780 \pm 0.027}$	$\mathbf{0.386 \pm 0.033}$

Table 3: Ablation experiment results on Dblp(ai) dataset. Bold indicates optimal results, underlined indicates sub-optimal results. ↑: higher is better, ↓: lower is better.

MIML algorithms and two state-of-the-art MGML methods. For fair comparison, the MIML baselines employ the transformation in Eq. (2) to convert graphs into instances, thereby reformulating the MGML task as an MIML problem while preserving structural information. PETAL learns bag-level representations in a self-supervised manner, which are then used as input to the RakelD classifier from scikit-learn for multi-label classification. Experiments employed ten-fold cross-validation, with average performance and standard deviations reported in Table 1. It is observed that PETAL consistently outperforms all MIML baselines across different datasets and evaluation metrics. This suggests that MIML methods are still insufficient for MGML tasks, even when structural information is preserved through the graph kernel. PETAL also surpasses all MGML baselines. This shows its strong ability in multi-label classification, even compared with supervised models that rely on extensive label annotations. Overall, these results demonstrate that the bag-level representations learned by PETAL are both informative and discriminative, which contributes to its superior performance on downstream MGML tasks.

Compared with Self-Supervised Baselines

We then validated the effectiveness of PETAL by comparing it with five advanced graph contrastive learning methods and one MIL algorithm as self-supervised baselines. For the graph contrastive learning baselines, max-pooling was employed to aggregate graph representations into the bag, whereas the MIL baseline utilized the transformation in Eq. (2) to convert graphs into instances. All methods learned bag-level representations in a self-supervised manner and were subsequently evaluated using the RakelD classifier from scikit-learn for downstream multi-label classification. The experiments were conducted using the reversed five-fold cross-validation (20% training, 80% testing), with results summarized in Table 2. It is observed that PETAL significantly outperforms all self-supervised baselines. This performance gap is because these baselines are designed for single-label scenarios and therefore have limited ability to capture label correlations. Moreover, they commonly employed data augmentation to generate positive samples for anchors, which can disrupt the semantic invariance assumption. In contrast, PETAL learns bag representations that preserve semantic invariance between anchors and positive samples while effectively modeling label correlations. As a result, PETAL achieves significant performance improvements over other SCL methods, demonstrating superior adaptability to downstream MGML tasks.

Ablation Analysis

To validate the effectiveness of PETAL's key components, we conducted comprehensive ablation studies on the Dblp(ai) dataset using the ten-fold cross-validation. For the multi-granularity contrastive loss, we removed: (1) the fine-grained graph-level loss (w/o $\mathcal{L}_1$), (2) the coarse-grained bag-level loss (w/o $\mathcal{L}_2$), and (3) the cross-granularity bag-graph loss (w/o $\mathcal{L}_3$). As shown in Table 3, the removal of the graph loss (w/o $\mathcal{L}_1$) resulted in the smallest decline, as the other losses partially retained fine-grained semantics. The removal of the bag loss (w/o $\mathcal{L}_2$) resulted in moderate degradation, underscoring its contribution to high-level label semantics. The removal of the bag-graph loss (w/o $\mathcal{L}_3$) resulted in the most significant performance drop, highlighting its role in capturing both local and global structures and semantic relations. For contrastive re-learning, we conducted an ablation by removing the re-learning strategy f_c (w/o f_c). As shown in Table 3, the removal of f_c significantly reduced performance due to semantic inconsistencies in positive pairs, indicating that f_c inherently constructs semantically consistent positive pairs without relying on additional data augmentations. Lastly, the removal of the label prototype $\boldsymbol{C}$ (w/o $\boldsymbol{C}$) led to substantial degradation, confirming its critical role in capturing label correlations and ensuring effective multi-label classification. Overall, these results highlight the importance of each component.

Conclusion

This work addresses the limitations of annotation dependency in MGML by proposing PETAL, a self-supervised contrastive re-learning framework. It performs dual-granularity feature-prototype alignment and leverages a multi-granularity contrastive loss to capture label correlations effectively. Through contrastive re-learning, PETAL generates augmentation-free positive samples, inherently ensuring semantic invariance without relying on data augmentation. These properties enable effective adaptation of SCL to MGML scenarios, such as computing power networks. Experiments on six benchmarks show that PETAL outperforms state-of-the-art supervised and self-supervised baselines by 4.12%. PETAL provides a practical solution for MGML applications where annotations are costly. Future work could investigate scalability for graph-level tasks.

Acknowledgments

This research acknowledges the generous support provided by the National Natural Science Foundation of China under Grant No. 62432003, and the Singapore Institute of Technology Ignition Grant under Grant No. R-IE3-A405-0005.

References

Asano, Y. M.; Rupprecht, C.; and Vedaldi, A. 2020. Self-Labelling via Simultaneous Clustering and Representation Learning. In *Proceedings of the 8th International Conference on Learning Representations*.

Bian, T.; Niu, Y.; Chang, H.; Yan, D.; Huang, J.; Rong, Y.; Xu, T.; Li, J.; and Cheng, H. 2024. Hierarchical Graph Latent Diffusion Model for Conditional Molecule Generation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 130–140. ACM.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep Clustering for Unsupervised Learning of Visual Features. In *Proceedings of the Computer Vision - ECCV 2018*, volume 11218 of *Lecture Notes in Computer Science*, 139–156. Springer.

Chen, X.; and He, K. 2021. Exploring Simple Siamese Representation Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 15750–15758. Computer Vision Foundation / IEEE.

Cuturi, M. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems*, 2292–2300.

Feng, X.; Zhang, H.; Zhang, Y.; Zhang, L. Y.; and Pan, S. 2025. BiMark: Unbiased Multilayer Watermarking for Large Language Models. *arXiv preprint arXiv:2506.21602*.

Feragen, A.; Kasenburg, N.; Petersen, J.; de Bruijne, M.; and Borgwardt, K. M. 2013. Scalable Kernels for Graphs with Continuous Attributes. In *Advances in Neural Information Processing Systems, United States*, 216–224.

Grill, J.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. Á.; Guo, Z.; Azar, M. G.; Piot, B.; Kavukcuoglu, K.; Munos, R.; and Valko, M. 2020. Bootstrap Your Own Latent-A New Approach to Self-Supervised Learning. In *Advances in Neural Information Processing Systems*.

Huang, M.; Zhao, Y.; Wang, Y.; Wahab, F.; Sun, Y.; and Chen, C. 2023. Multi-Graph Multi-Label Learning with Novel and Missing Labels. *Knowledge-Based Systems*, 276: 110753.

Huang, S.; Gao, W.; and Zhou, Z. 2019. Fast Multi-Instance Multi-Label Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(11): 2614–2627.

Li, Y.; Hu, J.; Jiang, Y.; and Zhou, Z. 2012. Towards Discovering What Patterns Trigger What Labels. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence, Toronto*, 1012–1018. AAAI Press.

Lin, N.; Qin, G.; Wang, G.; Zhou, D.; and Yang, A. 2023. An Effective Deployment of Contrastive Learning in Multi-label Text Classification. In *Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada*, 8730–8744. Association for Computational Linguistics.

Peng, C.; Wang, H.; Chen, K.; Shou, L.; Yao, C.; Wu, R.; and Chen, G. 2024. CoMAL: Contrastive Active Learning for Multi-Label Text Classification. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2364–2375. ACM.

Shervashidze, N.; Schweitzer, P.; van Leeuwen, E. J.; Mehlhorn, K.; and Borgwardt, K. M. 2011. Weisfeiler-Lehman Graph Kernels. *J. Mach. Learn. Res.*, 12: 2539–2561.

Sun, Q.; Wang, K.; Zhang, W.; Cheng, P.; and Lin, X. 2024. Interdependence-Adaptive Mutual Information Maximization for Graph Contrastive Learning. *IEEE Transactions on Knowledge and Data Engineering*, 36(12): 8556–8567.

Sun, Y.; and Fan, J. 2024. MMD Graph Kernel: Effective Metric Learning for Graphs via Maximum Mean Discrepancy. In *The Twelfth International Conference on Learning Representations, Vienna, Austria*. OpenReview.net.

Sun, Y.; and Kok, S. 2025. Unsupervised Multiple Kernel Learning for Graphs via Ordinality Preservation. In *The Thirteenth International Conference on Learning Representations, Singapore*. OpenReview.net.

Wang, M.; Zhao, Y.; Wang, Y.; Xu, T.; and Sun, Y. 2023a. Image Emotion Multi-Label Classification Based on Multi-Graph Learning. *Expert Systems With Applications*, 231: 120641.

Wang, Y.; Zhao, Y.; Wang, D. Z.; and Li, L. 2023b. GA-LOPA: Graph Transport Learning with Optimal Plan Alignment. In *Advances in Neural Information Processing Systems*.

Wang, Y.; Zhao, Y.; Wang, Z.; and Wang, M. 2023c. Robust Self-Supervised Multi-Instance Learning with Structure Awareness. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, 10218–10225. AAAI Press.

Wang, Y.; Zhao, Y.; Wang, Z.; Zhang, C.; and Wang, X. 2024. Robust Multi-Graph Multi-Label Learning with Dual-Granularity Labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10): 6509–6524.

Wu, J.; Pan, S.; Zhu, X.; Zhang, C.; and Wu, X. 2017. Positive and Unlabeled Multi-Graph Learning. *IEEE Transactions on Cybernetics*, 47(4): 818–829.

Wu, J.; Zhu, X.; Zhang, C.; and Yu, P. S. 2014. Bag Constrained Structure Pattern Mining for Multi-Graph Classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(10): 2382–2396.

Wu, Y.; Chen, Z.; Zou, Y.; Wang, S.; and Guo, W. 2025. Multi-Scale Structure-Guided Graph Generation for Multi-View Semi-Supervised Classification. *Expert System with Application*, 263: 125677.

Xia, J.; Wu, L.; Chen, J.; Hu, B.; and Li, S. Z. 2022. SimGRACE: A Simple Framework for Graph Contrastive Learning without Data Augmentation. In *Proceedings of the ACM Web Conference 2022*, 1070–1079. ACM.

Xiao, T.; Zhu, H.; Chen, Z.; and Wang, S. 2023. Simple and Asymmetric Graph Contrastive Learning without Augmentations. In *Advances in Neural Information Processing Systems, LA, USA*.

Xue, Z.; Yang, G.; Yu, X.; Yu, A.; Guo, Y.; Liu, B.; and Zhou, J. 2025. Multimodal Self-Supervised Learning for Remote Sensing Data Land Cover Classification. *Pattern Recognit.*, 157: 110959.

Yang, L.; Chen, X.; Zhuo, J.; Jin, D.; Wang, C.; Cao, X.; Wang, Z.; and Guo, Y. 2025. Disentangled Graph Spectral Domain Adaptation. In *Forty-second International Conference on Machine Learning*.

Yang, M.; Tang, W.; and Min, F. 2022. Multi-Instance Multi-Label Learning Based on Parallel Attention and Local Label Manifold Correlation. In *Proceedings of the International Conference on Data Science and Advanced Analytics, Shenzhen*, 1–10. IEEE.

You, Y.; Chen, T.; Shen, Y.; and Wang, Z. 2021. Graph Contrastive Learning Automated. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, 12121–12132. PMLR.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph Contrastive Learning with Augmentations. In *Advances in Neural Information Processing Systems*.

Zhang, H.; Wu, B.; Yang, X.; Yuan, X.; Liu, X.; and Yi, X. 2025. Dynamic Graph Unlearning: A General and Efficient Post-Processing Method via Gradient Transformation. In *WWW*, 931–944. ACM.

Zhang, H.; Yuan, X.; and Pan, S. 2024. Unraveling Privacy Risks of Individual Fairness in Graph Neural Networks. In *ICDE*, 1712–1725. IEEE.

Zhang, M.; and Zhou, Z. 2008. M3MIML: A Maximum Margin Method for Multi-Instance Multi-Label Learning. In *Proceedings of the 8th IEEE International Conference on Data Mining*, 688–697.

Zhang, M.; and Zhou, Z. 2014. A Review on Multi-Label Learning Algorithms. *IEEE Trans. Knowl. Data Eng.*, 26(8): 1819–1837.

Zhang, X.; Luo, T.; Liu, Y.; and Hou, C. 2024. Imbalanced Multi-Instance Multi-Label Learning via Coding Ensemble and Adaptive Thresholds. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 5413–5422. ACM.

Zhao, D.; Zhang, P.; Yin, H.; and Guo, J. 2025a. A Novel Multi-Layer Discriminative Dictionary Learning Approach for Image Classification. *Signal Processing*, 226: 109670.

Zhao, Y.; Wang, Y.; Wang, Z.; Shan, W.; Huang, M.; and Wang, X. 2025b. Graph Contrastive Learning with Progressive Augmentations. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Toronto*, 2079–2088. ACM.

Zhao, Y.; Wang, Y.; Wang, Z.; and Zhang, C. 2021. Multi-Graph Multi-Label Learning with Dual-Granularity Labeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*, 2327–2337. ACM.

Zhu, Z.; and Zhao, Y. 2018. Multi-Graph Multi-Label Learning Based on Entropy. *Entropy*, 20(4): 245.

Zhuo, J.; Liu, Y.; Lu, Y.; Ma, Z.; Fu, K.; Wang, C.; Guo, Y.; Wang, Z.; Cao, X.; and Yang, L. 2025a. DUALFormer: Dual Graph Transformer. In *The Thirteenth International Conference on Learning Representations*.

Zhuo, J.; Ma, Z.; Lu, Y.; Liu, Y.; Fu, K.; Jin, D.; Wang, C.; Wu, W.; Wang, Z.; Cao, X.; and Yang, L. 2025b. A Closer Look at Graph Transformers: Cross-Aggregation and Beyond. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.