Masked Graph Modeling with Multi-View Contrast

Yanchen Luo[1], Sihang Li[2], Yongduo Sui[1], Junkang Wu[1], Jiancan Wu[2], Xiang Wang[2*†]

MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition,
University of Science and Technology of China, Hefei, China
[1]{luoyanchen, syd2019, jkwu0909}@mail.ustc.edu.cn,
[2]{sihang0520, wujcan, xiangwang1223}@gmail.com

***Abstract*—Masked modeling has recently achieved remarkable success in specific fields of vision and language, sparking a surge of interest in graph-related research. However, Masked Graph Modeling (MGM), which captures fine-grained local information by masking low-level elements such as nodes, edges, and features, limits itself to a sub-optimal position, particularly on tasks requiring high-quality graph-level representations. Such a local perspective disregards the graph's global information and structure. To address these limitations, we propose a novel graph pre-training framework called Graph Contrastive Masked Autoencoder (GCMAE). GCMAE leverages the strengths of both MGM and Graph Contrastive Learning (GCL) to provide a more comprehensive perspective of both local and global. Our framework uses instance discrimination to learn global representations of graphs and reconstructs the graph using masked low-level elements. We augment the framework with a novel multi-view augmentation module to further enhance the pre-trained model's robustness and generalization ability. We evaluate GCMAE on real-world biochemistry and social network datasets, conducting extensive experiments on both node and graph classification tasks and transfer learning on downstream graph classification tasks. Our experimental results demonstrate that GCMAE's comprehensive perspective of both local and global benefits model pre-training. Moreover, GCMAE outperforms existing MGM and GCL baselines, proving its effectiveness on downstream tasks. Our code is available at https://github.com/lyc0930/GCMAE.**

***Index Terms*—Self-Supervised Learning, Masked Graph Modeling, Contrastive Learning, View Diversity**

I. Introduction

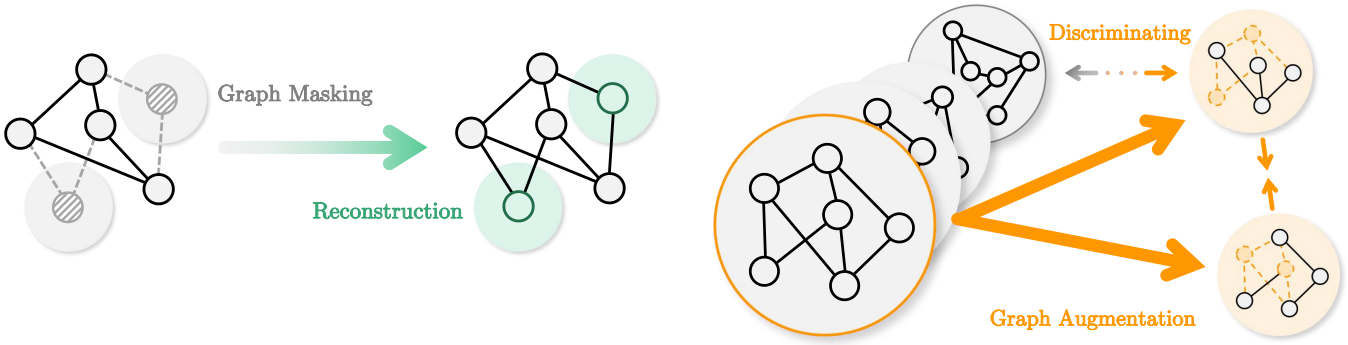
In recent years, learning high-transferability graph representations without manual supervision has become an important avenue of research, constituting the emerging field of graph pre-training [1], [2]. Among several approaches, graph self-supervised learning (GSSL) has risen to prominence. Inspired by the remarkable successes of masked modeling techniques in vision [3], [4], [5] and language [6], [7], masked graph modeling (MGM) [8], [9], [10], [11] is garnering interest from the GSSL community. MGM formulates the self-supervised learning task as a graph reconstruction problem. Specifically, it corrupts graph inputs by ablating low-level elements (*e.g.,* nodes [9], edges [12], or features [8]), and then optimizes models to reconstruct the missing elements, thereby capturing the low-level information. Once sufficiently pre-trained, the model can be fine-tuned on downstream tasks, leveraging the transferable representations to boost performance.

*Corresponding author
†Xiang Wang is also affiliated with Institute of Artificial Intelligence, Institute of Dataspace, Hefei Comprehensive National Science Center

Nevertheless, a limitation of current MGM methods is their focus on local graph structures without considering holistic graph semantics — reconstructing masked low-level elements encourages learning about local connectivity patterns, but fails to distinguish among different graphs. To our knowledge, an MGM paradigm that captures both local and global information is largely unexplored.

In this paper, we aim to bridge this gap by integrating graph contrastive learning (GCL) [13], [1], [14] into MGM. The key idea of GCL is to first create multiple augmented views of each graph by masking, then encourage augmentations of the same graph to cluster together, while pushing apart augmentations of different graphs. This forces the model to learn useful global representations that distinguish between graphs. While intuitively appealing, integrating GCL into MGM still faces potential challenges, which we summarize in two key questions:

- "*Would MGM benefit from contrastive learning?*" The divergent emphasis on graph information between MGM and GCL could potentially undermine their individual objectives. A simplistic combination of their goals may result in a compromise where MGM's reconstruction performance is sacrificed to accommodate GCL's graph differentiation. This potential conflict may explain why previous efforts have largely overlooked the incorporation of GCL into MGM.
- "*What kind of contrast will MGM benefit from?*" A common criticism of GCL is the heavy reliance on high-quality graph augmentation to create views [15], [16], [17]. Most GCL approaches utilize a single type of augmentation, such as node-level [1], [17], edge-level [18], [19], [20], [21], or feature-level masking [2], [22], [23]. This can risk corrupting the semantics of the graph and impairing generalization [24], [1], [14], [25]. For instance, masking certain functional groups in molecular graphs might alter their properties, thereby misleading the representation learning of the overall structure. Recent studies [26], [14], however, suggest that the contrast of multi-view can address this issue. Performing graph contrast across diverse views, such as simultaneous node and edge masking, can preserve the graph semantics from multiple perspectives, thus better preserving global information. Nonetheless, these findings are limited to GCL, and it remains an open question whether multi-view contrast can benefit MGM, given that most MGM methods use a single masking method [8], [9], [10], [11].



(a) Masked Graph Modeling (b) Graph Contrastive Learning

Fig. 1: Illustration of masked graph modeling (MGM) and graph contrastive learning (GCL). (a): MGM has a local perspective (*e.g.,* green area) on low-level elements (*i.e.,* node, edge and feature) in graphs by conducting substructure masking and reconstruction. (b): GCL has a global perspective (*e.g.,* yellow area) on graph-level representations by pulling the representations of two augmented views of the same anchor graph close and pushing those of different anchors away.

In this paper, we propose a novel approach to address these challenges. We present a unified framework that integrates MGM and GCL in a synergistic manner, leveraging the strengths of both while mitigating their potential conflicts. Furthermore, we introduce a multi-view contrast strategy to enhance the robustness and generalization of MGM. We demonstrate the effectiveness of our approach through comprehensive experiments, showing significant improvements over state-of-the-art methods. These two research questions motivate us to explore the potential of integrating MGM with GCL and introduce a simple but effective pre-training method, Graph Contrastive Masked Autoencoder (GCMAE), which synergistically combines the complementary strengths of MGM and GCL to learn comprehensive graph representations. By scrutinizing the constituent components of MGM and GCL, where MGM comprises graph masking and graph reconstruction while GCL involves graph augmentation and graph contrast, we identify a similar foundation, graph masking/augmentation, upon which to build. Therefore, we unify the first components of MGM and GCL, graph masking, by instantiating it as masking across node-, edge-, and feature-aware views. Building upon this joint masking, GCMAE contains two branches targeting local and global signals: graph reconstruction and graph contrast. The reconstruction branch employs a graph encoder-decoder on the feature view*, learning node representations to recover the masked node features and thereby distill the useful local information.

Meanwhile, the contrast branch guides the shared encoder through the implementation of multi-view contrastive learning. This learning approach consists of both intra-view and inter-view contrasts to distinguish between different graphs, thereby encoding global information. Crucially, the inter-view contrast serves as a bridge that connects the two branches by incorporating the feature view into the process of negative sampling, promoting the capture of multi-grained local and global signals. Empirical results show the effectiveness of GCMAE as compared to state-of-the-art MGM and GCL methods (*e.g.,* GraphMAE [8], RGCL [17]), in a wide range of graph learning tasks, including node classification, graph classification, and transfer learning [27], [28]. Overall, our contributions in this paper are summarized as:

- We introduce GCMAE, a novel framework that leverages contrastive learning to enhance graph representations learned via masked graph modeling.
- GCMAE provides a new multi-view perspective in graph representation learning, which can effectively intervene in the cross-information between multiple views, and further improve the model's capability to learn global information.
- Extensive experiments demonstrate that, on benchmark datasets (*e.g.,* Cora, CiteSeer, PubMed [29], and TU-Datasets [30]), GCMAE outperforms some leading methods (*e.g.,* GraphMAE [8] and RGCL [17]).

II. Related Work

Recent advances in self-supervised learning (SSL) on graphs have led to the emergence of two major approaches: generative and contrastive. The masked graph modeling (MGM) approach falls under the generative category, which masks nodes or edges in a graph, and then trains models to reconstruct the missing elements based on the remaining information. Graph contrastive learning (GCL) exemplifies the contrastive approach. It leverages the notion of similarity in embedding space, where similar instances are pulled closer and dissimilar ones are pushed apart. Both approaches have shown promising results in various graph-based tasks.

*We leave the study of other views as future work.

A. Masked Graph Modeling

In the realm of generative SSL frameworks, MGM has emerged as the leading approach for many graph-based tasks. Taking inspiration from the widely applied masked language modeling (MLM) and masked image modeling (MIM) techniques in language and vision fields, MGM pre-trains a graph autoencoder by masking a subset of low-level elements in the input graph and reconstructing the masked part with partially visible elements [6], [3].

Recent studies aimed to improve upon MGM, garnering significant attention. For instance, MGAE [10] proposes a tailored cross-correlation decoder with high-ratio edge masking to capture cross-correlations between masked edge endpoints at multiple granularities. GMAE [9] adopts masking and asymmetric encoding-decoding in graph transformers to reduce memory consumption. GraphMAE [8] focuses on node feature reconstruction with a scaled cosine error and re-masking decoding strategy. GraphMAE2 [31] regularizes feature reconstruction to improve robustness against disturbance in masked feature reconstruction. MaskGAE [12] employs vanilla edge masking and uses two decoders to predict the masked edges and the degrees of the associated nodes, respectively. HGMAE [11] applies MGM to heterogeneous graphs to handle various node attributes with different node positions.

However, current MGM frameworks suffer from a limitation in that their fine-grained substructure masking and reconstruction mechanism only provides local supervision signals and leaves the holistic graph semantics largely untouched, as illustrated in Figure 1(a). Consequently, MGM models may lack a global perspective and become sub-optimal, especially for graph-level tasks.

B. Graph Contrastive Learning

Graph contrastive learning (GCL) has emerged as an effective approach to obtaining instance-discriminative representations for graphs. The central idea behind GCL is to maximize agreement between differently augmented views of the same graph while minimizing agreement between views of different graphs. Divergent from MGM's focus on local structure, GCL optimizes the coarse-grained graph-level representations from a global perspective.

GraphCL [1] investigates the impact of various combinations of four types of graph augmentations on GCL. JOAO [32] formalizes GCL as a bi-level min-max optimization and enables learning to automatically select graph augmentation tasks and graph representation models. InfoGCL [33] reduces mutual information between contrasting components while maintaining the integrity of task-relevant information. AD-GCL [18] and RGCL [17] focus on preserving the semantics of the original graph in augmented views from the perspective of edge and node, respectively.

Compared to MGM, GCL provides supervision at the graph level, equipping the model with a global perspective. This raises the natural question of whether GCL could supplement MGM to gain comprehensive local and global features.

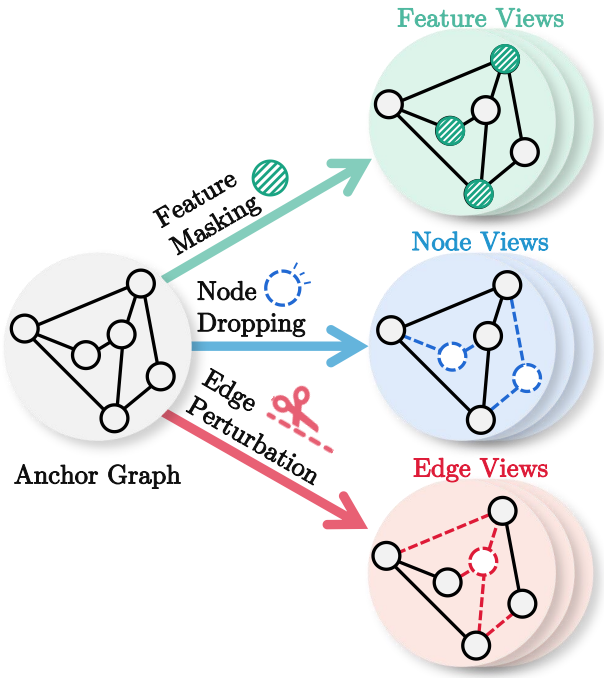


Fig. 2: Typical graph augmentation methods are operated on the low-level elements (nodes, edges, or features) of graphs including node dropping, edge perturbation, feature masking *etc.*which create different views

C. View Diversity

Graph augmentation is the prerequisite and crucial enabler for both MGM and GCL, as it masks or corrupts graph elements to create diverse views. Augmentations are typically operated on the nodes [13], [17], edges [18], [12], or features [2], [23], [22], [8], [31], as shown in Figure 2.

Conceptually, a high-quality graph augmentation is supposed to (1) preserve the semantic information (*i.e.,* label-invariant), and (2) contain an adequate level of diversity for better generalization. This dual importance of both properties has been highlighted in many GCL methods. SimCLR [24] points out that the combination of data augmentations plays a crucial role in effective prediction tasks and empirically shows that "no single transformation suffices to learn good representations". In graph area, GraphCL [1] demonstrates the crucial role of data augmentation in incorporating various priors for better representation learning, forming an assertion that "composing different augmentations benefits more". SRGCL [26] also verifies that an increase in view diversity can improve the performance of the pre-trained model.

The aforementioned GCL studies demonstrate that one single view (*i.e.,* only node-, edge- or feature-level augmentation) guides the pre-trained model to a sub-optimal position, while cross-view contrast benefits its representative ability and generalization performance. However, the potential of multi-view augmentation is under-explored in MGM, which uses single node [9], edge [10], [12] or feature [8], [31] masking. Specifi-

cally, GMAE [9] only employs a naive random node masking strategy that creates node views with large masking ratios to reduce the size of the input feature matrix, thereby alleviating the memory consumption associated with the transformer architecture. MaskGAE [12] attempts a masking strategy based on random walking to create edge views, only focusing on the masking and reconstruction of edges to construct supervision signals and reducing redundancy between paired subgraph views. GraphMAE [8] solely focuses on creating a feature view following the validated effectiveness of feature masking in CV and NLP.

Thus, we are motivated to explore the potential of view diversity in the combined design of MGM and GCL to further enhance the generalization.

III. Methodology

In this section, we present our novel graph pre-training framework, Graph Contrastive Masked Autoencoder (GCMAE). The overview of the proposed GCMAE is depicted in Figure 3, which consists of two branches: the reconstruction branch and the contrast branch. Now we introduce GCMAE following the pipeline of view generation, representation learning, and model optimization. We also provide the detailed algorithm of GCMAE for graph classification task in Algorithm 1 for your reference.

A. Notations

In this paper, we define $g = (\mathcal{V}, \mathcal{E}, \mathbf{X}) \in \mathcal{G}$ as a graph instance sampled from the graph set $\mathcal{G}$, with a node set $\mathcal{V}$ and an edge set $\mathcal{E}$. We use $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ to describe the node feature matrix, where $\mathbf{x}_i = \mathbf{X}[i,:]$ is the d-dimensional feature vector of node $v_i \in \mathcal{V}$. As shown in Figure 3, our GCMAE contains three distinct modules. Specifically, we define f_E as the GNN encoder to be pre-trained. f_D and f_P are denoted as the GNN decoder and projection head, respectively. In graph pre-training tasks, we first pre-train the backbone model f_θ, and then apply the pre-trained model to the downstream tasks. We define the representation matrix as $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times d_h}$, where each row $\mathbf{h}_i = \mathbf{H}[i,:]$ denotes the feature representation of node v_i, and d_h denotes the dimension of the latent space. For the downstream tasks, our primary focus is centered on node and graph classification tasks. In graph classification, we can use $\mathrm{Readout}(\cdot)$ function (*e.g.,* average pooling) to summarize node representations to a graph-level representation.

B. View Generation

Generally, graph masking, which we consider to be a prerequisite of both MGM and GCL, masks or corrupts a subset of low-level elements (*e.g.,* node set $\mathcal{V}$ [9], edge set $\mathcal{E}$ [12] or node feature matrix $\mathbf{X}$ [8]) in the anchor graph to create its augmented views for the subsequent reconstruction or contrast. As a general framework, the graph masking strategy in our proposed GCMAE is not confined and can be instantiated as any of the aforementioned ones at different granularities.

Feature View. Inspired by its proven success in prior MGM studies [8], [31], we adopt the feature masking and tokenization technique in the reconstruction branch of GCMAE. This approach enables the backbone encoder to capture the information of low-level node features. Specifically, a subset of nodes $\tilde{\mathcal{V}} \subset \mathcal{V}$ is randomly sampled from the node set $\mathcal{V}$ under the constraint of a fixed masking ratio ρ_m (*i.e.,* $|\tilde{\mathcal{V}}| = \rho_m |\mathcal{V}|$) For each node $v_i \in \tilde{\mathcal{V}}$, we replace its feature with a special trainable token (*i.e.,* mask token) $\mathbf{x}_{[\text{mask}]} \in \mathbb{R}^d$. Accordingly, the node vector $\tilde{\mathbf{x}}_i$ for $v_i \in \mathcal{V}$ in the masked feature matrix $\tilde{\mathbf{X}}$ can be obtained as:

$$\tilde{\mathbf{x}}_i = \begin{cases} \mathbf{x}_{[\text{mask}]}, & v_i \in \tilde{\mathcal{V}} \\ \mathbf{x}_i, & v_i \notin \tilde{\mathcal{V}} \end{cases}. \tag{1}$$

Without disturbing the original topological structure of the graph (*i.e.,* keeping the edge set $\mathcal{E}$ unchanged), the feature view is generated:

$$g_m = (\mathcal{V}, \mathcal{E}, \tilde{\mathbf{X}}). \tag{2}$$

And the goal of the reconstruction branch is to recover the original graph $g = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ based on this partially observed feature view g_m.

Node and Edge View. To construct the node view, we adopt node masking. Specifically, we randomly mask a subset of nodes $\tilde{\mathcal{V}}$ from the set of node $\mathcal{V}$, under the constraint of a fixed node masking ratio ρ_n as well. With the un-masked nodes $\mathcal{V} \setminus \tilde{\mathcal{V}}$ identified, the edges $\mathcal{E}_n$ between them are preserved to construct the node view:

$$g_n = (\mathcal{V} \setminus \tilde{\mathcal{V}}, \mathcal{E}_n, \mathbf{X}_n). \tag{3}$$

Analogically, edge view can be generated from the anchor graph by masking a certain ratio ρ_e of edges and then removing isolated nodes:

$$g_e = (\mathcal{V}_e, \mathcal{E} \setminus \tilde{\mathcal{E}}, \mathbf{X}_e). \tag{4}$$

As illustrated in Figure 3, the feature, node, and edge views comprise elements of varying granularities. This diversity allows them to serve as complementary sources of information, enriching the perspective provided to the backbone encoder. Thereafter, they will be utilized in different self-supervised tasks — masked node feature reconstruction and multi-view contrastive learning — to empower the encoder with more comprehensive global perspectives.

C. Representation Learning and Optimization

Upon acquiring the feature, node, and edge views of each anchor graph, we feed them into a shared GNN [34], [35] encoder f_{E} (*i.e.,* the backbone model to be pre-trained), thereby producing corresponding representations for the contrast and reconstruction branches.

1) Reconstruction Branch:

Based on the principle that the feature of each node can be implicitly inferred by its neighboring nodes in GNNs [36], we try to reconstruct the masked node features based on the partially observed feature view $g_m = (\mathcal{V}, \mathcal{E}, \tilde{\mathbf{X}})$.

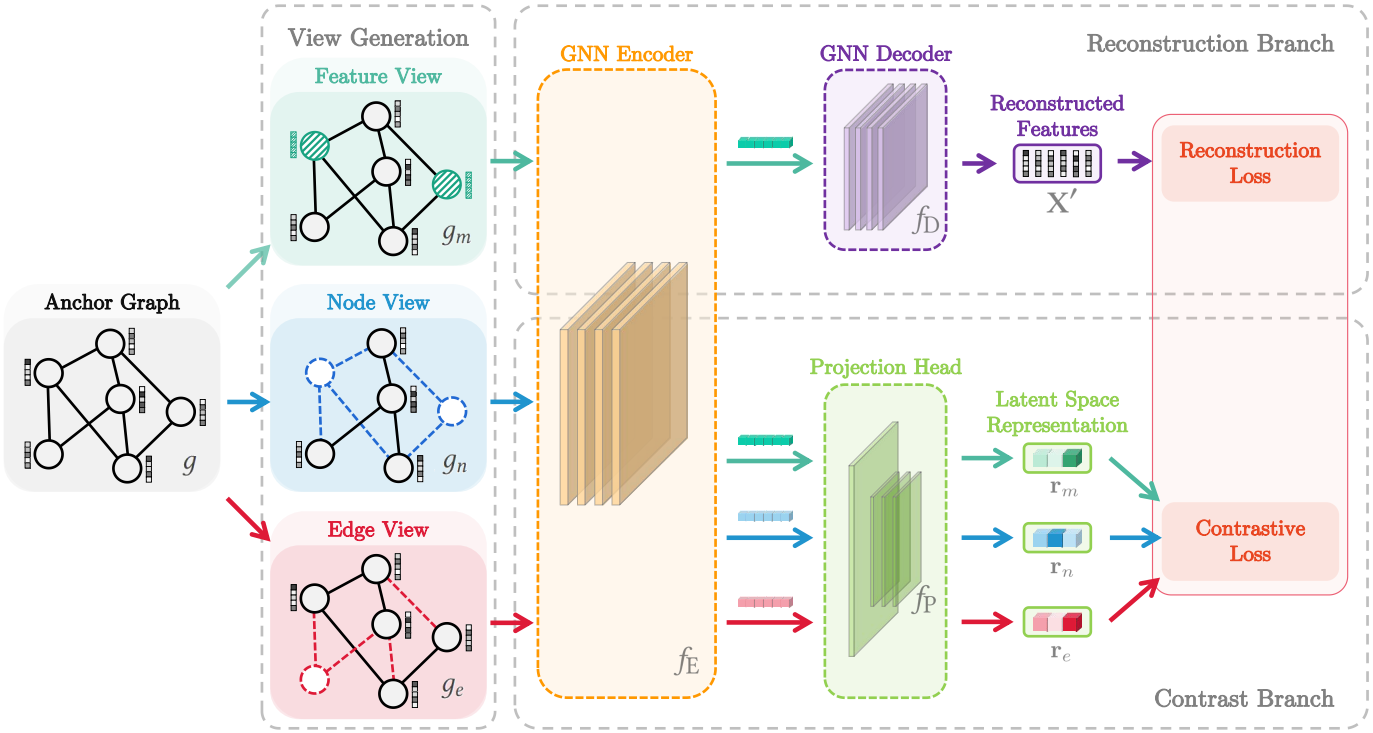


Fig. 3: Framework of GCMAE. Our framework contains three major components: view generation, reconstruction branch, and contrast branch. Specifically, given an anchor graph g, we generate diverse views of feature g_m, node g_n and edge g_e. Then augmented views are fed into a shared graph neural network (GNN) encoder f_{E} being pre-trained. In the reconstruction branch, the GNN decoder f_{D} learns to reconstruct the masked node features based on the representations of feature view g_m provided by the encoder f_{E}. In the contrast branch, we first obtain the representations of g_m, g_n, g_e through the encoder, and then use a multilayer perceptron (MLP)-based projection head f_{P} to project these representations to a latent hypersphere space and conduct contrastive learning. After pre-training, only the encoder f_{E} is kept for the downstream tasks.

Encoding. For a given graph g, the backbone GNN encoder f_{E} takes in its feature view g_m and generates its representations $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times d_h}$ with a hidden dimension of d_h for each node:

$$\mathbf{H} = f_{\mathrm{E}}(g_m). \tag{5}$$

Re-mask Decoding. A decoder f_{D} attempts to map the representations $\mathbf{H}$ back to the original feature matrix $\mathbf{X}$. Here we follow the re-mask decoding technique as in GraphMAE [8] to force the encoder to learn more condensed representations. Specifically, we replace the hidden representation of the masked nodes with another re-mask token $\mathbf{h}_{[\text{re-mask}]} \in \mathbb{R}^{d_h}$ to obtain the re-masked representation $\tilde{\mathbf{H}}$, in which vector $\tilde{\mathbf{h}}_i$ corresponding to node v_i is formulated as:

$$\tilde{\mathbf{h}}_i = \begin{cases} \mathbf{h}_{[\text{re-mask}]}, & v_i \in \tilde{\mathcal{V}} \\ \mathbf{h}_i, & v_i \notin \tilde{\mathcal{V}} \end{cases}, \tag{6}$$

where $\tilde{\mathcal{V}}$ is the same subset as in Section III-B. Then, the decoder f_{D} gives the reconstructed output $\mathbf{X}' \in \mathbb{R}^{|\mathcal{V}| \times d}$ as:

$$\mathbf{X}' = f_{\mathrm{D}}(\tilde{g}), \quad \tilde{g} = (\mathcal{V}, \mathcal{E}, \tilde{\mathbf{H}}). \tag{7}$$

Reconstruction Loss. To balance the contribution of samples with different difficulties during training, we use the scaled cosine error [8] as the reconstruction loss:

$$l_{\mathrm{rec}}(g) = \frac{1}{|\tilde{\mathcal{V}}|} \sum_{v_i \in \tilde{\mathcal{V}}} \left(1 - \frac{\mathbf{x}_i^{\mathsf{T}} \mathbf{x}_i'}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_i'\|}\right)^{\gamma}, \tag{8}$$

where $\tilde{\mathcal{V}} \subset \mathcal{V}$ is the set of masked node in graph g, $\mathbf{x}_i$ and $\mathbf{x}_i'$ are the original and reconstructed features of node v_i, respectively. $gamma$ is a scaling hyperparameter adjusted for each dataset.

Along with the re-masking decoding, this loss enables the reconstruction branch to perform masked modeling on each graph, encouraging the encoder to capture graph information from the local perspective.

2) Contrast Branch:

We build the contrast branch upon the node view g_n, edge view g_e and feature view g_m (See Section III-B). Specifically, we feed these views into the GNN backbone encoder f_{E} and a graph readout (*i.e.,* pooling) function $\mathrm{Readout}(\cdot)$ to obtain

the graph representations:

$$\begin{aligned}\mathbf{z}_n &= \text{Readout}(f_{\text{E}}(g_n)),\\ \mathbf{z}_e &= \text{Readout}(f_{\text{E}}(g_e)),\\ \mathbf{z}_m &= \text{Readout}(f_{\text{E}}(g_m)).\end{aligned} \tag{9}$$

Henceforth, we follow the commonly adopted operation of projecting them into a latent hypersphere space through an MLP-based projection head f_{P} with l_2 normalization:

$$\begin{aligned}\mathbf{r}_n &= f_{\text{P}}(\mathbf{z}_n),\\ \mathbf{r}_e &= f_{\text{P}}(\mathbf{z}_e),\\ \mathbf{r}_m &= f_{\text{P}}(\mathbf{z}_m).\end{aligned} \tag{10}$$

Contrastive Loss. To maximize the mutual information among node, edge, and feature views of the same anchor graph, we introduce multi-view contrast into the Info-NCE loss [37]. For the node view in this contrast, we not only consider negative samples from the intra-view but also treat the feature view that retains the complete graph structure as negative samples from the inter-view.

$$l_{\text{con}}(g) = -\log \frac{\exp\left({\mathbf{r}_n}^{\top}\mathbf{r}_e/\tau_1\right)}{\exp\left({\mathbf{r}_n}^{\top}\mathbf{r}_e/\tau_1\right) + N_{\text{intra}} + N_{\text{inter}}} \tag{11}$$

where N_{intra} and N_{inter} denotes the distinct contribution of negative samples within the intra-view and across the inter-view, respectively:

$$\begin{aligned}N_{\text{intra}} &= \sum_{\mathbf{r}_n^- \in g'^-} \exp\left({\mathbf{r}_n}^{\top}\mathbf{r}_n^-/\tau_1\right),\\ N_{\text{inter}} &= \sum_{\boldsymbol{z_f}^- \in g'^-} \exp\left({\mathbf{r}_n}^{\top}\mathbf{r}_m^-/\tau_2\right)\end{aligned}, \tag{12}$$

and τ_1 and τ_2 are two adaptive hyperparameters, which are set differently by initial parameters and asynchronously adjusted during the training process based on the contrastive loss to selectively regulate the model's learning granularity for each graph instance.

It's worth mentioning that in the node classification task, intra-view negative sample $\mathbf{r}_n^-$ and inter-view negative sample $\mathbf{r}_m^-$ is implicitly provided by other views generated in the same minibatch. As for the graph classification task, negative samples come from views of other graphs in the same minibatch with g, where $g'^- \in \mathcal{G}^-$ summarizes the representations of the node or edge views of the other graphs.

Adap-τ. To achieve adaptive fine-grained temperature control for each graph instance, we employ the Adap-τ technique from [38]. This technique allows the contrastive loss to carefully consider the contribution of negative samples from both intra-view and inter-view perspectives.

$$\tau^* = \tau_0 \cdot \exp\left(\mathbb{W}\left(\max\left(-\frac{1}{e}, \frac{l_{\text{con}}(g) - m}{2\beta}\right)\right)\right), \tag{13}$$

where $\mathbb{W}(\cdot)$ stands for the Lambert-W function, m serves as a threshold to identify hard samples based on their respective loss, and β is a hyperparameter to regulate the temperature and avoid gradient vanishing.

Algorithm 1 GCMAE for Graph Classification

Initialization: dataset $\mathcal{G}$, GNN encoder $f_{\text{E}}(\cdot)$, GNN decoder $f_{\text{D}}(\cdot)$, projection head $f_{\text{P}}(\cdot)$, sampling ratio ρ_n & ρ_e, temperature τ and tradeoff hyperparameter λ.
for Sampled minibatch $\mathcal{G}_b = \{g_i : i = 1, 2, ..., N\} \subset \mathcal{G}$ **do**
  for graph instance $g = (\mathcal{V}, \mathcal{E}, \mathbf{X}) \in \mathcal{G}_b$ **do**
    # View Generation
    Randomly sample nodes $\tilde{\mathcal{V}} \subset \mathcal{V}$ and edges $\tilde{\mathcal{E}} \subset \mathcal{E}$.
    Feature view $g_m = (\mathcal{V}, \mathcal{E}, \tilde{\mathbf{X}})$, $\tilde{\mathbf{X}} \xleftarrow{\text{mask}} \mathbf{X}$ ▷ (1)
    Node view $g_n = (\mathcal{V} \setminus \tilde{\mathcal{V}}, \mathcal{E}_n, \mathbf{X}_n)$
    Edge view $g_e = (\mathcal{V}_e, \mathcal{E} \setminus \tilde{\mathcal{E}}, \mathbf{X}_e)$
    # Reconstruction Branch
    $\mathbf{H} = f_{\text{E}}(g_m)$ ▷ (5)
    $\tilde{g} = (\mathcal{V}, \mathcal{E}, \tilde{\mathbf{H}})$, $\tilde{\mathbf{H}} \xleftarrow{\text{re-mask}} \mathbf{H}$ ▷ (6)
    $\mathbf{X}' = f_{\text{D}}(\tilde{g})$ ▷ (7)
    $l_{\text{rec}}(g) = \frac{1}{|\tilde{\mathcal{V}}|} \sum_{v_i \in \tilde{\mathcal{V}}} \left(1 - \frac{\mathbf{x}_i^{\top}\mathbf{x}_i'}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_i'\|}\right)^{\gamma}$ ▷ (8)
    # Contrast Branch
    $\mathbf{z}_m = \text{Readout}(f_{\text{E}}(g_m))$, ▷ (9)
    $\mathbf{z}_n = \text{Readout}(f_{\text{E}}(g_n))$, $\mathbf{z}_e = \text{Readout}(f_{\text{E}}(g_e))$
    $\mathbf{r}_m = f_{\text{P}}(\mathbf{z}_m)$, $\mathbf{r}_n = f_{\text{P}}(\mathbf{z}_n)$, $\mathbf{r}_e = f_{\text{P}}(\mathbf{z}_e)$ ▷ (10)
    $N_{\text{intra}} = \sum_{\mathbf{r}_n^- \in g'^-} \exp\left({\mathbf{r}_n}^{\top}\mathbf{r}_n^-/\tau_1\right)$, ▷ (12), (13)
    $N_{\text{inter}} = \sum_{\boldsymbol{z_f}^- \in g'^-} \exp\left({\mathbf{r}_n}^{\top}\mathbf{r}_m^-/\tau_2\right)$
    $l_{\text{con}}(g) = -\log \frac{\exp\left({\mathbf{r}_n}^{\top}\mathbf{r}_e/\tau_1\right)}{\exp\left({\mathbf{r}_n}^{\top}\mathbf{r}_e/\tau_1\right) + N_{\text{intra}} + N_{\text{inter}}}$ ▷ (11)
  end for
  $\mathcal{L} = \frac{1}{N} \sum_{g \in \mathcal{G}_b} \left(l_{\text{rec}}(g) + \lambda \cdot l_{\text{con}}(g)\right)$ ▷ (14)
  Update f_{E}, f_{D}, f_{P} to minimize $\mathcal{L}$.
end for
return GNN encoder f_{E}

The contrast branch receives the node, edge, and feature views as input, utilizing cross-view and cross-instance contrast to capture the global information of the anchor graph, thereby distinguishing itself from other instances. By endowing the backbone encoder with a holistic and global instance-discrimination capability, the contrast branch enhances its performance. Moreover, through the introduction of the feature view that preserves structure as an inter-view negative sample in the loss, the contrastive branch has the potential to further maximize mutual information among multiple views and establish a tighter connection with the reconstruction branch.

3) Overall Training Objective:

The reconstruction branch captures relevant information by masking low-level node features, while the contrast branch captures a graph's global information distinguishable from other graphs. Hence, to further unleash the power of GCL on MGM, we integrate these two branches and define our overall training objective as a weighted combination of reconstruction loss (See Eq. 8) and contrastive loss (See Eq. 11):

$$\min_{f_{\mathrm{E}}, f_{\mathrm{D}}, f_{\mathrm{P}}} \mathcal{L} = \mathbb{E}_{g \in \mathcal{G}}[l_{\mathrm{rec}}(g) + \lambda \cdot l_{\mathrm{con}}(g)], \tag{14}$$

where λ is the hyperparameter controlling the tradeoff between the reconstruction and Contrastive Loss. By optimizing this overall optimization objective, we train the encoder to acquire the ability to capture both local and global information of the graph input and generate high-quality representations. After pre-training, the decoder f_{D} and the projection head f_{P} are discarded, while only the encoder f_{E} equipped with both local and global perspective is preserved for downstream tasks.

It is worth mentioning that our GCMAE is a model-agnostic and scalable self-supervised graph learning framework, which applies to different backbone graph models. The vanilla versions of the reconstruction and contrast branches described in this section can be further improved by employing new variants of the MGM and GCL frameworks as reviewed in Section II for better local and global information capture.

Conceptually, an ideal pre-trained graph backbone model is supposed to be able to handle various downstream tasks, which can be roughly categorized into node-level, link-level (*i.e.,* edge-level), or graph-level, with varying levels of granularity across low-level and graph-level entities. Thus, we argue that the ability to generate high-quality representations of both fine-grain (*i.e.,* node- and edge-level) and coarse-grain (*i.e.,* graph-level) is a must that comes from the local and global perspectives of the model, respectively. Convention MGM and GCL frameworks are confined to a single perspective of either local or global, which results in suboptimal performance and restricts their usage across diverse tasks. However, our proposed hybrid framework, GCMAE tackles this problem by equipping the pre-trained model with a more comprehensive perspective and better generalization ability to node-level, edge-level, and graph-level tasks.

IV. Experiments

In this section, extensive experiments are conducted on datasets of bio-chemistry and social networks, with unsupervised node-level, edge-level and graph-level classification settings, to demonstrate the effectiveness of GCMAE. Empirical results show that GCMAE, as a general and scalable self-supervised framework, outperforms existing competitive graph pre-training baselines and sets the new state-of-the-art on various benchmark tasks.

A. Node Classification

1) Setup and baselines:

We conduct experiments on 6 benchmark node classification datasets: Cora, CiteSeer, PubMed [29], Ogbn-arxiv [39], Computer and Reddit, to validate the effectiveness of GCMAE. In particular, the testing for Reddit follows the inductive setup in GraphSage [40]. The detailed statistics of datasets are included in Table I. To make a fair comparison, the same evaluation protocol and backbone model structure is adopted as in previous baselines [12], [8]. Specifically, a standard GAT-based [34] graph encoder is pre-trained with GCMAE, whose parameters are then frozen to generate the representations of all nodes. But it should be pointed out that our framework allows various choices of the encoder architecture, such as GIN [41], GAT [34] and GraphSage [40], without constraints. Hereafter, a linear classifier is trained and we report the mean and variance of accuracy on the test nodes with 10 times of random initialization. GCMAE is then compared with competitive baselines including both generative (GAE [42], MaskGAE [12], and GraphMAE [8]) and contrastive (DGI [43], GMI [44], GRACE [20], GCA [19], MVGRL [14], BGRL [21] and SUGRL [45]) self-supervised frameworks.

2) Performance of GCMAE.:

The overall results are showcased in Table I, and the following observations can be obtained:

- **MGM surpasses GCL on node-level tasks.** Node classification is a typical task focusing on low-level elements (*i.e.,* nodes), which requires the backbone encoder to be empowered with a local perspective of capturing the information of substructures. As pointed out in our motivation, compared with GCL, MGM's inherent mechanism of graph property (*i.e.,* node, edge or feature) masking and reconstruction makes the model pre-trained by it more suitable for node-level tasks. Table I shows that MGM-based frameworks (*e.g.,* MaskGAE and GraphMAE) achieve better performance than GCL-based ones (*e.g.,* DGI, GMI, GRACE, GCA, MVGRL, BGRL and SUGRL), which empirically verifies aforementioned statement.
- **GCMAE outperforms existing self-supervised baselines.** The reconstruction branch in GCMAE equipped the backbone encoder with a local perspective as other MGM schemes and its low-level representative ability is further enhanced as the contrast branch cooperates and provides supplementary holistic information. Thus, GCMAE achieves the best performance on 4 out of 6 datasets and 2 second best compared with leading self-supervised generative and contrastive schemes. Meanwhile, as a self-supervised scheme, it outperforms models trained in supervised manners on 4 datasets and achieves comparable results on the others, demonstrating its promising performance of generating high-quality node-level representations.

B. Graph Classification

1) Setup and baselines:

Furthermore, we follow the settings in InfoGraph [46] to evaluate GCMAE in the unsupervised graph-level representation learning, which contains 6 benchmark datasets from

TABLE I: Experiment results in unsupervised representation learning for node classification. Test accuracies (%) on multiple academic and social network datasets. Statistics are from their original papers. **Boldface** and underline indicate the best and the second best performance on each dataset, respectively.

	Dataset	Cora	CiteSeer	PubMed	Obgn-arxiv	Computer	Reddit
Statistics	#Nodes	2,708	3,327	19,717	169,343	13,752	232,965
	#Edges	10,556	9,104	88,648	2,315,598	491,722	11,606,919
	#Features	1,433	3,703	500	128	767	602
	#Classes	7	6	3	40	10	41
Supervised	GCN[1]	81.5 ±0.2	70.3 ±0.4	79.0 ±0.5	71.74±0.29	86.51±0.54	95.3 ±0.1
	GAT[1]	83.0 ±0.7	72.5 ±0.7	79.0 ±0.3	72.10±0.13	86.93±0.29	96.0 ±0.1
Unsupervised	GAE	71.5 ±0.4	65.8 ±0.4	72.1 ±0.5	63.60±0.50	86.66±0.07	-
	DGI	82.3 ±0.6	71.8 ±0.7	76.8 ±0.6	65.10±0.40	83.95±0.47	94.0 ±0.1
	GMI[1]	83.00±0.30	72.4 ±0.1	79.9 ±0.2	68.20±0.20	82.21±0.31	94.9 ±0.02
	GRACE[1]	81.90±0.40	71.20±0.50	80.60±0.40	68.70±0.40	86.25±0.25	94.2 ±0.0
	GCA[1]	81.80±0.20	71.90±0.40	81.00±0.30	68.20±0.20	87.85±0.31	-
	MVGRL[1]	82.90±0.30	72.60±0.40	80.10±0.70	68.10±0.10	86.90±0.10	-
	BGRL[1]	82.86±0.49	71.41±0.92	82.05±0.85	71.64±0.12	**90.34±0.19**	94.22±0.03
	SUGRL	83.40±0.50	73.00±0.40	81.90±0.30	69.30±0.20	88.90±0.20	-
	MaskGAE	84.05±0.18	73.49±0.59	83.06±0.22	70.73±0.26	89.51±0.08	-
	GraphMAE	84.2 ±0.4	73.4 ±0.4	81.1 ±0.4	71.75±0.17	-	**96.01±0.08**
	GraphMAE2	84.5 ±0.6	73.4 ±0.3	81.4 ±0.5	-	-	-
	GCMAE	**85.13±0.67**	**73.60±0.44**	**83.21±0.54**	**72.08±0.23**	89.58±0.20	95.91±0.10

[1] The reported results are partly from MaskGAE [12].

[30]: NCI1, PROTEINS, MUTAG, COLLAB, REDDIT-B and IMDB-B, covering both biochemical and social network domain. A GIN-based [41] is pre-trained with GCMAE and the encoded graph representations are then fed into a non-linear SVM classifier to evaluate the discriminative quality of those representations. We compare GCMAE with previous self-supervised learning baselines: untrained GIN [41], InfoGraph [46], GraphCL [1], JOAO [32], AD-GCL [16], RGCL [17] and GraphMAE [8]. The summaries of datasets and test accuracies over 10 random initializations are reported in Table II.

2) Performance of GCMAE:

Table II showcases the performance of GCMAE and other baselines, and we get the following observations:

- **GCL outperforms MGM on graph-level tasks.** In contrast to node-level tasks requiring the local perspective, the global perspective of encoding discriminative semantic information is crucial for the classification of graph-level tasks. The variants of GCL (*e.g.,* InfoGraph [46], GraphCL [1], JOAO [32], AD-GCL [16] and RGCL [17]), with the shared goal of instance-discrimination while ignoring the fine-grained representations of node, edge or features, are empowered with this global perspective, making it more suitable than the MGM frameworks (*e.g.,* GraphMAE [8]) for graph-level tasks. This observation is consistent with our motivations to leverage GCL in MGM to provide an additional global perspective and the capability of graph-level discrimination.
- **GCMAE outperforms existing self-supervised baselines.** The contrast branch in GCMAE enables the backbone encoder to generate instance-discriminative graph representations (*i.e.,* global perspective). Meanwhile, node representation is the foundation of that of a graph, because the graph-level representation of a graph instance is transformed from its node-level representation by a pooling layer. So the

TABLE II: Experiment results in unsupervised representation learning for graph classification. Test accuracies (%) on multiple biochemical and social network datasets. Statistics are from their original papers except GraphMAE. **Boldface** and underline indicate the best and the second best performance on each dataset, respectively.

Dataset	NCI1	PROTEINS	MUTAG	COLLAB	REDDIT-B	IMDB-B	Avg.	Gain
#Graphs	4,110	1,113	188	5,000	2,000	1,000		
#Avg. Nodes	29.87	39.06	17.93	74.49	429.63	19.77		
#Avg. Edges	32.30	72.82	19.79	2457.78	497.75	96.53		
No Pre-Train	65.40±0.17	72.73±0.51	87.39±1.09	65.29±0.16	76.86±0.25	69.37±0.37	72.84	-
graph2vec	73.22±1.81	73.30±2.05	83.15±9.25	-	75.28±1.03	71.10±0.54	-	-
InfoGraph	76.20±1.06	74.44±0.31	**89.01±1.13**	70.05±1.13	82.50±1.42	73.03±0.87	77.54	4.70
GraphCL	77.87±0.41	74.39±0.45	86.80±1.34	71.36±1.15	89.53±0.74	71.14±0.44	78.52	5.68
JOAO	78.36±0.53	74.07±1.10	87.67±0.79	69.33±0.34	86.42±1.45	70.83±0.25	77.78	4.94
AD-GCL	75.86±0.62	75.04±0.48	88.62±1.27	74.89±0.90	**92.35±0.42**	71.49±0.98	79.71	6.87
RGCL	78.14±1.08	75.03±0.43	87.66±1.01	70.92±0.65	90.34±0.58	71.85±0.84	78.99	6.15
GraphMAE[1]	76.12±1.45	74.69±0.74	87.21±0.88	73.16±2.48	87.76±1.08	71.61±0.52	75.33	5.18
GCMAE	**78.56±1.35**	**76.84±0.93**	88.24±1.01	**76.97±1.13**	88.25±1.12	**74.59±0.81**	**80.56**	**7.72**

[1] GraphMAE is reproduced since its evaluation metrics are different from the other baselines.

reconstruction branch implicitly enhanced the pre-trained model on graph-level tasks by improving its node-level representative ability. This is empirically verified in Table II, where GCMAE achieves the best performance on 4 out of 6 datasets compared with existing MGM- and GCL-based schemes, reaches the highest average test accuracy of 77.03% and shows its potential as a general and scalable graph pre-training framework.

C. Transfer Learning

1) Setup and baselines:

Following the commonly adopted transfer learning settings and evaluation metrics in [2], the backbone model is pre-trained on a large label-free dataset – ZINC-2M [47] which includes 2 million unlabeled molecules sampled from the ZINC15 and fine-tuned on 8 downstream datasets with graph-level classification tasks in MoleculeNet [48] to evaluate the transferability of schemes. The detailed statistics of datasets are in Table IV and the downstream datasets are split by scaffold-split to mimic real-world use cases GCMAE is compared with competitive graph pre-training baselines, including Infomax [43], EdgePred [49], AttrMasking [2], ContextPred [2], GraphCL [1], GraphLoG [50], AD-GCL [16], JOAO [32], RGCL [17] and GraphMAE [8]. To make a fair comparison, the same evaluation protocol and backbone model structure are adopted as in previous baselines [8]. Specifically, a standard GIN-based [41] graph encoder is pre-trained with GCMAE, whose parameters are then frozen to generate the representations of all nodes. But it should be pointed out that our framework allows various choices of the encoder architecture, such as GIN [41], GAT [34] and GraphSage [40], without constraints. The ROC-AUC scores over 10 random initializations are reported in Table III

2) Performance of GCMAE:

Table III showcases the performance of GCMAE and other baselines. GCMAE achieves the best performance on 3 out of 8 datasets and the highest average gain compared with existing MGM- and GCL-based schemes. The performance of GCMAE on molecular property prediction in chemistry and protein function prediction in biology, which pre-trains and finetunes the model in different datasets shows the transferability of the proposed pre-training scheme. Compared to MGM-based schemes GraphMAE [8], GCMAE's performance demonstrates a notable improvement in the generalization and transferability of MGM with multi-view contrast in graph representation learning which allows the pre-trained model to capture discriminative features in downstream datasets.

TABLE III: Experiment results in transfer learning on downstream graph classification tasks. ROC-AUC scores (%) on molecular property prediction benchmarks. Statistics are from their original papers. **Boldface** and underline indicate the best and the second best performance on each dataset, respectively.

Dataset	BBBP	Tox21	ToxCast	SIDER	ClinTox	MUV	HIV	BACE	AVG.	GAIN
No Pre-Train	65.8 ±4.5	74.0 ±0.8	63.4 ±0.6	57.3 ±1.6	58.0 ±4.4	71.8 ±2.5	75.3 ±1.9	70.1 ±5.4	67.0	-
Infomax	68.8 ±0.8	75.3 ±0.5	62.7 ±0.4	58.4 ±0.8	69.9 ±3.0	75.3 ±2.5	76.0 ±0.7	75.9 ±1.6	70.3	3.3
EdgePred	67.3 ±2.4	76.0 ±0.6	64.1 ±0.6	60.4 ±0.7	64.1 ±3.7	74.1 ±2.1	76.3 ±1.0	79.6 ±1.2	70.3	3.3
AttrMasking	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	4.1
ContextPred	68.0 ±2.0	75.7 ±0.7	63.9 ±0.6	60.9 ±0.6	65.9 ±3.8	75.8 ±1.7	77.3 ±1.0	79.6 ±1.2	70.9	3.9
GraphCL	69.68±0.67	73.87±0.66	62.40±0.57	60.53±0.88	75.99±2.65	69.80±2.66	**78.47±1.22**	75.38±1.44	70.77	3.77
GraphLoG	**72.5 ±0.8**	75.7 ±0.5	63.5 ±0.7	61.2 ±1.1	76.7 ±3.3	76.0 ±1.1	77.8 ±0.8	**83.5 ±1.2**	73.4	6.4
AD-GCL	70.01±1.07	76.54±0.82	63.07±0.72	**63.28±0.79**	79.78±3.52	72.30±1.61	78.28±0.97	78.51±0.80	72.72	5.72
JOAO	70.22±0.98	74.98±0.29	62.94±0.48	59.97±0.79	81.32±2.49	71.66±1.43	76.73±1.23	77.34±0.48	71.90	4.9
RGCL	71.42±0.66	75.20±0.34	63.33±0.17	61.38±0.61	**83.38±0.91**	76.66±0.99	77.90±0.80	76.03±0.77	73.16	6.16
GraphMAE	72.0 ±0.6	75.5 ±0.6	64.1 ±0.3	60.3 ±1.1	82.3 ±1.2	76.3 ±2.4	77.2 ±1.0	83.1 ±0.9	73.8	6.8
GCMAE	72.11±0.85	**77.24±0.35**	**65.22±0.26**	61.40±1.75	80.76±1.40	**77.79±1.34**	78.32±0.51	81.44±0.87	74.29	7.29

TABLE IV: Statistics for ZINC and MoleculeNet.

Datasets	Graphs#	Avg. N#	Avg. E#
ZINC-2M	2,000,000	26.62	57.72
BBBP	2,039	24.06	51.90
Tox21	7,831	18.57	38.58
ToxCast	8,576	18.78	38.52
SIDER	1,427	33.64	70.71
ClinTox	1,477	26.15	55.76
MUV	93,087	24.23	52.55
HIV	41,127	25.51	54.93
BACE	1,513	34.08	73.71

D. Link Prediction

1) Setup and baselines:

We further evaluate the performance of GCMAE on link prediction tasks, which is a typical downstream task in graph representation learning. Following the settings in [12], we conduct experiments on 3 benchmark datasets: Cora, CiteSeer, PubMed, removing 5% of edges for the validation set and 10% for the test. The detailed statistics of datasets can be found in Table I. We compare GCMAE with competitive graph pre-training baselines, including GAE [42], VGAE [42], ARGA [51], ARVGA [51], SAGE [40], MGAE [10], GraphMAE [8] and MaskGAE [12]. The AUC and average precision (AP) scores over 10 random initializations are reported in Table V.

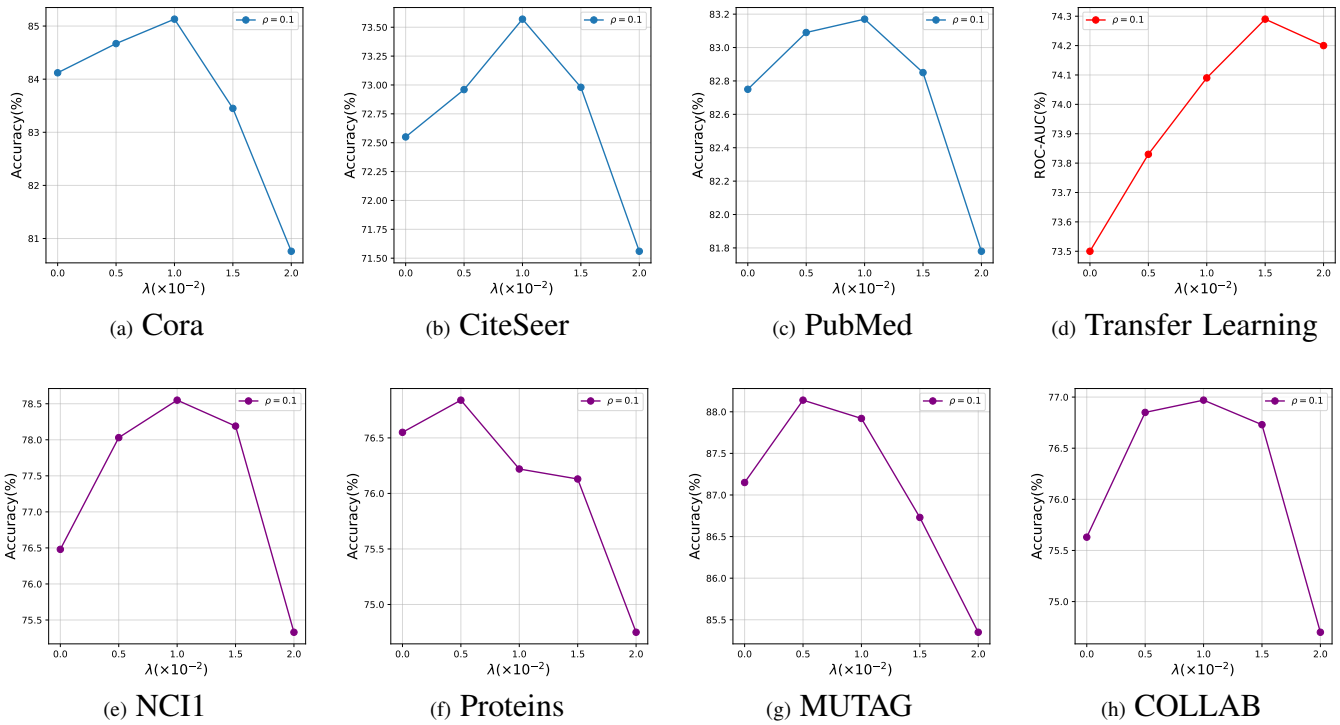
2) Performance of GCMAE:

Table V showcases the performance of GCMAE and other baselines. The performance of GCMAE on link prediction achieves a notable performance over the compared baselines on both 2 metrics. Specifically, GCMAE achieves the best performance on 5 out of 6 evaluations, which demonstrates the effectiveness of GCMAE on link prediction tasks. The result demonstrates that the graph reconstruction capability from the MGM benefits from the GCL, which helps the self-supervised learning of GCMAE on link prediction.

E. Hyperparameter Sensitivity

To gain insight into the impact of reconstruction and contrast branches on model performance and validate the effectiveness of our framework design, we conduct a series of experiments on the hyperparameter sensitivity.

Component analysis. As shown in Table I and Table II, on both node and graph classification tasks, GCMAE outperforms the variants of pure MGM and GCL schemes, demonstrating the effectiveness of our combination design of reconstruction and contrast branches. Further, we evaluate its sensitivity to the trade-off parameter λ in Equation (14) on 3 node classification datasets (Cora, CiteSeer, PubMed [29]) and 4 graph classification datasets [30]: (NCI1, PROTEINS, MUTAG, COLLAB). Moreover, we do the analysis of the components based on the average ROC-AUC value for transfer learning experiments on



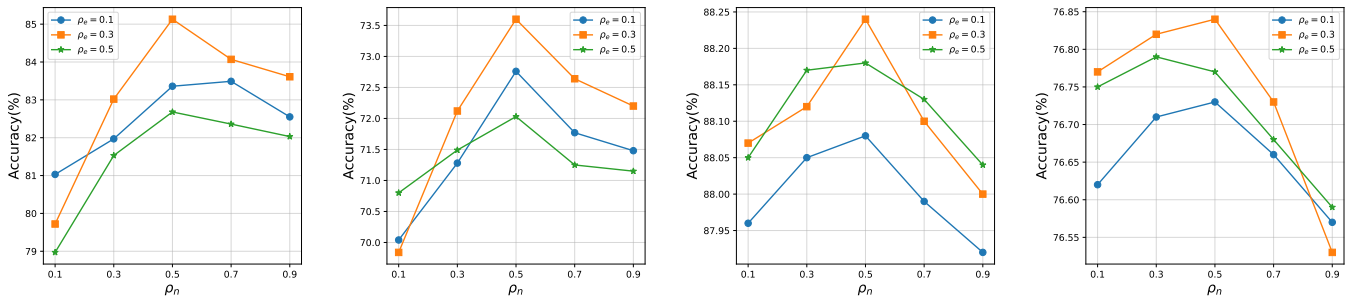
(a) Cora (b) CiteSeer (c) PubMed (d) Transfer Learning

(e) NCI1 (f) Proteins (g) MUTAG (h) COLLAB

Fig. 4: Impact of the trade-off parameter λ. The upper left (blue), the upper right (red) and the lower (purple) column show the sensitivity of the model performance to the trade-off parameter λ on 3 node classification datasets, transfer learning downstream tasks average and 4 graph classification datasets, respectively.

downstream tasks. In Figure 4, we show the change of model performance *w.r.t.* to λ. Affected by the different training loss function scales, on the magnitude of 10^{-2}, we observe that the model reaches the pike performance when λ is around 0.01, on all curves of node classification datasets and 2 graph classification tasks (NCI1, COLLAB) and reaches the pike when λ is around 0.005 on the other 2 datasets. And when λ is larger than the pike point, the model performance markedly deteriorates across all datasets of varying scales, indicating that despite the enhancements brought about by divergent perspectives at different granularities, merely comparing two distinct views is insufficient for learning. This aligns with the observations made in MVGRL [14]. Nevertheless, concerning transfer learning, when λ exceeds the peak, the model's performance in downstream tasks does not significantly diminish, underscoring the ability of contrastive learning to effectively ensure the quality of graph representations learned in transfer learning, thereby corroborating our standpoint. When the feature reconstruction loss dominates (*i.e.,* λ approaches 0), GCMAE simply degenerates into vanilla MGM disregarding global information and shows comparable performance with GraphMAE [8]. It can be further corroborated by the slight variation in the model performance between node classification and graph classification tasks: as the model loses its capability of instance discriminating due to the absence of multi-view contrast, its performance on graph classification tasks undergoes a more pronounced decline.

Mask ratio analysis. We also analyze the effect of mask ratio ρ_n, ρ_e on model performance on node classification and graph classification datasets: Cora, CiteSeer, MUTAG and Proteins, as Figure 5 shows. A low mask ratio of node/feature (*i.e.,* ρ_n) harms the model performance, mainly because the feature reconstruction task under a low mask ratio is too simple for encoders to learn high-quality and robust node representations, which is consistent with [8]. Meanwhile, when the mask ratio increases above 0.5, the model performance decreases as well. Intuitively, when the majority of nodes are masked out, the encoder-decoder lacks adequate local information to infer neighboring nodes and the graph contrast is also difficult to discriminate due to the constraints of global information. It's worth mentioning that compared with the nodes/feature mask ratio, the edge mask ratio is generally lower due to the relatively fluctuating effect of edge masks on the structure of views. In all 4 datasets, an excessively high or low mask ratio of either node view or edge view significantly reduces the model performance. The reduction of the model performance caused by the higher mask ratio of node and feature view is significantly larger in the 2 graph classification datasets than in the node classification tasks which indicates that an



(a) Cora (b) CiteSeer (c) MUTAG (d) Proteins

Fig. 5: Impact of mask ratio ρ_n and ρ_e. The effect of mask ratio of node/feature ρ_n and edge ρ_e on model performance on 2 node classification datasets and 2 graph classification datasets. Each figure shows the variation of model performance *w.r.t.* ρ_n, for three different values of ρ_e.

TABLE V: Experiment results in link prediction. AUC and average precision (AP) scores (%) on multiple citation networks datasets. **Boldface** and underline indicate the best and the second best performance on each dataset, respectively.

Dataset	Cora		CiteSeer		PubMed	
	AUC	AP	AUC	AP	AUC	AP
GAE[1]	91.09±0.01	92.83±0.03	90.52±0.04	91.68±0.05	96.40±0.01	96.50±0.02
VGAE[1]	91.40±0.01	92.60±0.01	90.80±0.02	92.00±0.02	94.40±0.02	94.70±0.02
ARGA[1]	92.40±0.00	93.23±0.00	91.94±0.00	93.03±0.00	96.81±0.00	97.11±0.00
ARVGA[1]	92.40±0.00	92.60±0.00	92.40±0.00	93.00±0.00	96.50±0.00	96.80±0.00
SAGE[1]	86.33±1.06	88.24±0.87	85.65±2.56	87.90±2.54	89.22±0.87	89.44±0.82
MGAE[1]	95.05±0.76	94.50±0.86	94.85±0.49	94.68±0.34	98.45±0.03	98.22±0.05
GraphMAE	87.62±0.88	89.30±0.76	86.90±1.20	88.90±1.10	90.80±0.70	91.00±0.70
MaskGAE[1]	96.66±0.17	96.29±0.23	97.95±0.10	98.12±0.10	**99.06±0.05**	98.99±0.06
GCMAE	**96.85±0.34**	**97.04±0.56**	**98.52±0.35**	**99.02±0.19**	99.04±0.74	**99.12±0.48**

[1] The reported results are partly from MaskGAE [12].

excessively high node and feature mask imposes a pronounced constraint on the discrimination of global information. This result shows that in the MGM framework enhanced by GCL, the masking of low-level elements in the graph plays a crucial role, thus we suggest tuning ρ_n and ρ_e carefully.

V. Conclusion

In this work, we introduce a novel graph self-supervised learning framework, Graph Contrastive Masked Autoencoder (GCMAE), which aims to enhance Masked Graph Modeling (MGM) with Graph Contrastive Learning (GCL) from a multi-view perspective. In GCMAE, diverse views at the different granularity of node, edge and feature are generated for reconstruction and contrast branches. Extensive experiments on node classification, graph classification tasks and transfer learning on downstream tasks show that GCMAE significantly improves the representative and generalization ability of the pre-trained model. In the future, we plan to extend GCMAE to more tasks such as transfer learning to verify its capability of representation learning, and to replace the backbone model with novel architectures such as Graphormer [52] or Graph-Trans [53] to further tap the potential of MGM and GCL and establish a more general graph pre-training scheme.

VI. Acknowledgement

This research is supported by the National Natural Science Foundation of China (9227010114, 62302321) and the University Synergy Innovation Program of Anhui Province (GXXT-2022-040).

REFERENCES

[1] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *NeurIPS*, 2020.

[2] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. S. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," in *ICLR*, 2020.

[3] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick, "Masked autoencoders are scalable vision learners," in *CVPR*, 2022, pp. 15 979–15 988.

[4] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, "Simmim: a simple framework for masked image modeling," in *CVPR*. IEEE, 2022, pp. 9643–9653.

[5] M. Assran, M. Caron, I. Misra, P. Bojanowski, F. Bordes, P. Vincent, A. Joulin, M. Rabbat, and N. Ballas, "Masked siamese networks for label-efficient learning," in *ECCV (31)*, 2022, pp. 456–473.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT (1)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[8] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, and J. Tang, "Graphmae: Self-supervised masked graph autoencoders," in *KDD*, 2022, pp. 594–604.

[9] H. Chen, S. Zhang, and G. Xu, "Graph masked autoencoder," *arXiv preprint arXiv:2202.08391*, 2022.

[10] Q. Tan, N. Liu, X. Huang, R. Chen, S. Choi, and X. Hu, "MGAE: masked autoencoders for self-supervised learning on graphs," *CoRR*, vol. abs/2201.02534, 2022.

[11] Y. Tian, K. Dong, C. Zhang, C. Zhang, and N. V. Chawla, "Heterogeneous graph masked autoencoders," *CoRR*, vol. abs/2208.09957, 2022.

[12] J. Li, R. Wu, W. Sun, L. Chen, S. Tian, L. Zhu, C. Meng, Z. Zheng, and W. Wang, "Maskgae: Masked graph modeling meets graph autoencoders," *CoRR*, vol. abs/2205.10053, 2022.

[13] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "GCC: graph contrastive coding for graph neural network pre-training," in *KDD*, 2020, pp. 1150–1160.

[14] K. Hassani and A. H. K. Ahmadi, "Contrastive multi-view representation learning on graphs," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 119, 2020, pp. 4116–4126.

[15] H. Zhang, Q. Wu, J. Yan, D. Wipf, and P. S. Yu, "From canonical correlation analysis to self-supervised graph neural networks," in *NeurIPS*, 2021, pp. 76–89.

[16] S. Suresh, P. Li, C. Hao, and J. Neville, "Adversarial graph augmentation to improve graph contrastive learning," in *NeurIPS*, 2021, pp. 15 920–15 933.

[17] S. Li, X. Wang, A. Zhang, Y. Wu, X. He, and T. Chua, "Let invariant rationale discovery inspire graph contrastive learning," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022, pp. 13 052–13 065.

[18] S. Suresh, P. Li, C. Hao, and J. Neville, "Adversarial graph augmentation to improve graph contrastive learning," in *NeurIPS*, 2021, pp. 15 920–15 933.

[19] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *WWW*, 2021, pp. 2069–2080.

[20] ——, "Deep graph contrastive representation learning," *CoRR*, vol. abs/2006.04131, 2020.

[21] S. Thakoor, C. Tallec, M. G. Azar, R. Munos, P. Velickovic, and M. Valko, "Bootstrapped representation learning on graphs," *CoRR*, vol. abs/2102.06514, 2021.

[22] Z. Hu, Y. Dong, K. Wang, K. Chang, and Y. Sun, "GPT-GNN: generative pre-training of graph neural networks," in *KDD*, 2020, pp. 1857–1867.

[23] A. Salehi and H. Davulcu, "Graph attention auto-encoders," in *ICTAI*, 2020, pp. 989–996.

[24] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, ser. Proceedings of Machine Learning Research, 2020, pp. 1597–1607.

[25] N. Lee, J. Lee, and C. Park, "Augmentation-free self-supervised learning on graphs," in *AAAI*. AAAI Press, 2022, pp. 7372–7380.

[26] Anonymous, "Self-attentive rationalization for graph contrastive learning," in *Submitted to The Eleventh International Conference on Learning Representations*, 2023, under review. [Online]. Available: https://openreview.net/forum?id=CdU7ApBxICO

[27] Z. Yang, W. W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 48, 2016, pp. 40–48.

[28] P. Yanardag and S. V. N. Vishwanathan, "Deep graph kernels," in *KDD*. ACM, 2015, pp. 1365–1374.

[29] Z. Yang, W. W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 48. JMLR.org, 2016, pp. 40–48.

[30] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "Tudataset: A collection of benchmark datasets for learning with graphs," in *ICMLW*, 2020.

[31] Z. Hou, Y. He, Y. Cen, X. Liu, Y. Dong, E. Kharlamov, and J. Tang, "Graphmae2: A decoding-enhanced masked self-supervised graph learner," in *WWW*. ACM, 2023, pp. 737–746.

[32] Y. You, T. Chen, Y. Shen, and Z. Wang, "Graph contrastive learning automated," in *ICML*, 2021, pp. 12 121–12 132.

[33] D. Xu, W. Cheng, D. Luo, H. Chen, and X. Zhang, "Infogcl: Information-aware graph contrastive learning," in *NeurIPS*, 2021, pp. 30 414–30 425.

[34] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR (Poster)*. OpenReview.net, 2018.

[35] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR (Poster)*. OpenReview.net, 2017.

[36] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 1263–1272.

[37] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.

[38] J. Chen, J. Wu, J. Wu, S. Zhou, X. Cao, and X. He, "Adap-tau: Adaptively modulating embedding magnitude for recommendation," *CoRR*, vol. abs/2302.04775, 2023.

[39] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," in *NeurIPS*, 2020.

[40] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS*, 2017, pp. 1024–1034.

[41] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2019.

[42] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *CoRR*, vol. abs/1611.07308, 2016.

[43] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *ICLR (Poster)*, 2019.

[44] Z. Peng, W. Huang, M. Luo, Q. Zheng, Y. Rong, T. Xu, and J. Huang, "Graph representation learning via graphical mutual information maximization," in *WWW*. ACM / IW3C2, 2020, pp. 259–270.

[45] Y. Mo, L. Peng, J. Xu, X. Shi, and X. Zhu, "Simple unsupervised graph representation learning," in *AAAI*. AAAI Press, 2022, pp. 7797–7805.

[46] F. Sun, J. Hoffmann, V. Verma, and J. Tang, "InfoGraph: unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *ICLR*, 2020.

[47] T. Sterling and J. J. Irwin, "Zinc 15–ligand discovery for everyone," *Journal of chemical information and modeling*, vol. 55, no. 11, pp. 2324–2337, 2015.

[48] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "Moleculenet: a benchmark for molecular machine learning," *Chemical science*, vol. 9, no. 2, pp. 513–530, 2018.

[49] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS*, 2017, pp. 1024–1034.

[50] M. Xu, H. Wang, B. Ni, H. Guo, and J. Tang, "Self-supervised graph-level representation learning with local and global structure." *Cornell University - arXiv,Cornell University - arXiv*, Jun 2021.

[51] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *IJCAI*. ijcai.org, 2018, pp. 2609–2615.

[52] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T. Liu, "Do transformers really perform badly for graph representation?" in *NeurIPS*, 2021, pp. 28 877–28 888.

[53] Z. Wu, P. Jain, M. A. Wright, A. Mirhoseini, J. E. Gonzalez, and I. Stoica, "Representing long-range context for graph neural networks with global attention," in *NeurIPS*, 2021, pp. 13 266–13 279.