Structure-Aware Hard Negative Mining for Heterogeneous Graph Contrastive Learning

Yanqiao Zhu[1,2,†], Yichen Xu[3,†], Hejie Cui[4], Carl Yang[4], Qiang Liu[1,2], and Shu Wu[1,2,‡]
[1]Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
yanqiao.zhu@cripac.ia.ac.cn, {qiang.liu, shu.wu}@nlpr.ia.ac.cn
[3]School of Computer Science, Beijing University of Posts and Telecommunications
linyxus@bupt.edu.cn
[4]Department of Computer Science, Emory University
{hejie.cui, j.carlyang}@emory.edu

***Abstract*—Recently, Graph Neural Networks (GNNs) have been widely used to model and analyze Heterogeneous Graphs (HGs), while most of them rely on a relatively large amount of labeled data. Contrastive Learning (CL), a key approach in self-supervised learning, is a promising direction to alleviate the label scarcity problem in training heterogeneous GNNs. In this work, we investigate CL on HGs and propose a novel method dubbed HeterOgeneous gRAph Contrastive learning with structure-aware hard nEgative mining, HORACE for brevity. At first, we generate multiple semantic views for HGs based on different metapaths. Unlike most multiview CL methods that maximizes the consistency among different views, we propose a novel multiview contrastive aggregation objective that adaptively learns information from each semantic view. Moreover, considering the complex graph structure and the smoothing nature of GNNs, we propose a structure-aware hard negative mining scheme that measures hardness by structural characteristics for HGs. By synthesizing more negative nodes, we upweight hard negatives with limited computational overhead to further boost the performance. Empirical studies on three real-world datasets show that our proposed method consistently outperforms existing state-of-the-art methods and notably, even surpasses several supervised counterparts.**

***Index Terms*—Heterogeneous information networks, graph contrastive learning, structure-aware hard negative mining, graph neural networks**

I. Introduction

Many real-world complex interactive objectives can be represented in Heterogeneous Graphs (HGs) or heterogeneous information networks. Recent development in heterogeneous Graph Neural Networks (GNNs) has achieved great success in analyzing heterogeneous structure data [1, 2]. However, most existing models require a relatively large amount of labeled data for proper training [3–6], which may not be accessible in reality. As a promising strategy of leveraging abundant unlabeled data, Contrastive Learning (CL), as a case of self-supervised learning, is proposed to learn representations by distinguishing semantically similar samples (positives) over dissimilar samples (negatives) in the latent space. Most existing CL methods follow a multiview paradigm, where they construct multiple views of the input data via identity-preserving augmentations [7] and maximize consistency of representations among these views. Though multiview CL has achieved promising performance in many tasks [8–12], we argue that it is still non-trivial to adopt multiview CL on HG data.

At first, since multiple types of nodes and edges convey abundant semantic information, it is straightforward to construct views based on HG semantics such as metapaths. Following the multiview contrastive objective, its embeddings in different semantic views constitute positives and all other embeddings are regarded as negative examples. However, this scheme fails to consider the inter-view dependency of different semantic views (e.g., complementary or redundant information [13]) and may lead to suboptimal performance. For example, consider an academic network, where nodes correspond to four types of entities: papers (P), conferences (C), topics (T), and authors (A). Two semantic views created by APA and APCPA share *common* co-authorship information, while two other metapaths APCPA and APTPA connect authors from two *dissimilar* sources: conferences and topics. Therefore, it is insufficient to distill comprehensive information from HGs by only contrasting node representations within each semantic view.

Secondly, the previous scheme assumes that all negative samples make equal contribution to the CL objective. Previous research in computer vision [14–16] has established that the *hard negative sample* is of particular concern for effective CL. To be specific, the more similar a negative sample to its anchor, the more helpful it is for learning effective representatives. When dealing with HGs, due to the neighborhood aggregation scheme in each semantic view [3], heterogeneous GNN produces similar embeddings within ego networks; embeddings of neighboring nodes sharing *the same label* with the anchor node thus tend to be similar to the anchor. Therefore, how to appropriately select hard negatives to further benefit CL for HGs remains rarely explored.

To address the aforementioned issues, in this paper we propose HeterOgeneous gRAph Contrastive learning with structure-aware hard nEgative mining, HORACE for brevity, as shown in Figure 3. The HORACE works by constructing multiple semantic views from the HG at first. Then, we learn node embeddings within each semantic view and combine them

[†] The first two authors contributed equally to this work.
[‡] To whom correspondence should be addressed.

into an aggregated representation. Thereafter, we propose a novel multiview contrastive aggregation objective for HG data, whose aim is to ensure global consistency among semantic views and thus adaptively encode information from each view. Finally, regarding hard negative sampling for HGCL, instead of measuring hardness of nodes using similarity between node representations, we propose to discover hard negatives from *structural aspects*. In particular, we measure hardness of each negative pair according to structural characteristics and synthesize more negatives by randomly mixing up these selected negatives, so as to give larger weights to harder negatives. The proposed structure-aware scheme enriches the selection of negatives with structure embeddings, which yields harder negative samples in the context of HGs. Measuring hardness through structural characteristics enjoys another benefit that being irrespective of training progress, which can be used in supplement to poor representations in the initial training stage.

In summary, the main contribution of this work is twofold.

- We propose a novel CL framework that enables self-supervised training for HGs from both semantic and structural aspects. Specifically, we propose a novel contrastive aggregation objective that adaptively learn information from each semantic view. Also, we propose to enrich the contrastive objective with structurally hard negatives to further improve the performance.
- Extensive experiments on three real-word datasets from various domains demonstrate the effectiveness of the proposed method. Particularly, our HORACE method outperforms representative unsupervised baseline methods, achieves competitive performance with supervised counterparts, and even exceeds some of them.

II. Preliminaries

A. Problem Definition

We introduce several key definitions of heterogeneous graphs and the problem of unsupervised heterogeneous graph representation learning.

Definition II.1 (Heterogeneous graph)**.** A heterogeneous graph (HG), denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X}, \boldsymbol{R}, \phi, \varphi)$, is a graph with multiple types of nodes and edges, where $\mathcal{V}, \mathcal{E}$ denote the node set and the edge set respectively. The node type mapping function $\phi : \mathcal{V} \to \mathcal{S}$ associates each node $v_i \in \mathcal{V}$ with a node type $s = \phi(v_i)$, the edge type mapping function $\varphi : \mathcal{E} \to \mathcal{R}$ associates each edge $e_{ij} \in \mathcal{E}$ with an edge type $r = \varphi(e_{ij})$, with $|\mathcal{S}| + |\mathcal{R}| > 2$. Moreover, each node v_i and each edge e_{ij} is possibly associated with attribute $\boldsymbol{x}_i^o$ and $\boldsymbol{r}_{ij}^r$. Note that the edge type $r = \varphi(e_{ij})$ implicitly defines types of its two end nodes v_i and v_j.

Definition II.2 (Metapath)**.** A metapath p defines a path on the network schema in the form of $s_1 \xrightarrow{r_1} s_2 \xrightarrow{r_2} \cdots \xrightarrow{r_l} s_{l+1}$. It represents a composite relation $r_1 \circ r_2 \circ \cdots \circ r_l$ between two nodes v_1 and v_{l+1} that captures the proximity between the two nodes from a particular semantic perspective, where $\circ$ is the composite operator. We further denote the set of all considered metapaths as $\mathcal{P}$.

Definition II.3 (Heterogeneous graph representation learning)**.** Given a HG $\mathcal{G}$, the problem of heterogeneous graph representation learning aims to learn node representations $\boldsymbol{H} \in \mathbb{R}^{|\mathcal{V}| \times d}$ that encode both structural and semantic information, where $d \ll |\mathcal{V}|$ is the dimension of the embedding space.

B. Heterogeneous Graph Neural Networks

Most heterogeneous GNN [3, 6] learns node representations under different semantic views and then aggregates them using attention networks. Following their approaches, we first generate multiple semantic views, each corresponding to one metapath that encodes one aspect of semantic information. Then, we leverage an attentive network to compute semantic-specific embedding $\boldsymbol{h}_i^p$ for node v_i under metapath p as

$$\boldsymbol{h}_i^p = \Big\Vert_{k=1}^{K} \sigma \left(\sum_{v_j \in \mathcal{N}_p(v_i)} \alpha_{ij}^p \boldsymbol{W}^p \boldsymbol{x}_j \right), \tag{1}$$

where $\|$ concatenates K standalone node representations in each attention head, $\mathcal{N}_r(v_i)$ defines the neighborhood of v_i that is connected by metapath p, $\boldsymbol{W}^p \in \mathbb{R}^{d \times m}$ is a linear transformation matrix for metapath p, and $\sigma(\cdot)$ is the activation function, such as $\operatorname{ReLU}(\cdot) = \max(0, \cdot)$. The attention coefficient α_{ij}^p can be computed by a softmax function

$$\alpha_{ij}^p = \frac{\exp(\sigma(\boldsymbol{a}_p^\top [\boldsymbol{h}_i^p \parallel \boldsymbol{h}_j^p]))}{\sum_{v_k \in \mathcal{N}_p(v_i)} \exp(\sigma(\boldsymbol{a}_p^\top [\boldsymbol{h}_i^p \parallel \boldsymbol{h}_k^p]))}, \tag{2}$$

where $\boldsymbol{a}_p \in \mathbb{R}^{2d}$ is a trainable semantic-specific linear weight vector.

Finally, we combine node representation in each view to an aggregated representation. We employ another attentive network to obtain the semantic-aggregated representation $\boldsymbol{h}_i$ that combines information from every semantic space by

$$\boldsymbol{h}_i = \sum_{p=1}^{|\mathcal{P}|} \beta^p \boldsymbol{h}_i^p. \tag{3}$$

The coefficients are given by

$$\beta^p = \frac{\exp(w^p)}{\sum_{p' \in \mathcal{P}} \exp(w^{p'})}, \tag{4}$$

$$w^p = \frac{1}{|\mathcal{V}|} \sum_{v_i \in \mathcal{V}} \boldsymbol{q}^\top \cdot \tanh(\boldsymbol{W} \boldsymbol{h}_i^p + \boldsymbol{b}), \tag{5}$$

where $\boldsymbol{q} \in \mathbb{R}^{d_m}$ is the semantic-aggregation attention vector, $\boldsymbol{W} \in \mathbb{R}^{d_m \times d}, \boldsymbol{b} \in \mathbb{R}^{d_m}$ is the weight matrix and the bias vector respectively, and d_m is a hyperparameter.

III. Motivating Studies

In this section, by analyzing a real-world dataset, we identify the deficiency of applying the vanilla multiview contrastive objective for HGs and provide an intuition of the proposed contrastive aggregation objectives. Following that, we elaborate the necessity of introducing structure-aware hard negative mining.

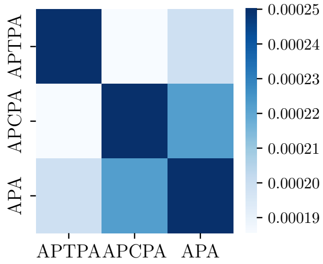


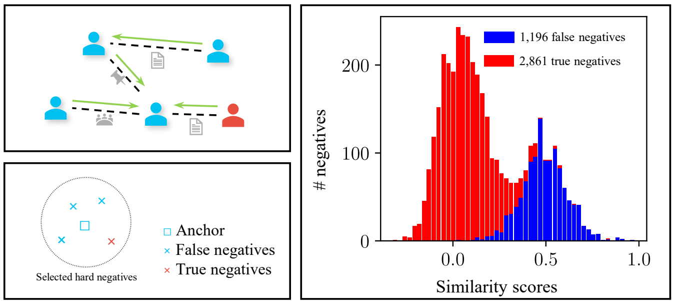
Fig. 1: Node-averaged mutual information of authors' embeddings between every semantic view pairs.

A. Intuition of Contrastive Aggregation Objectives for HGs

When we generate semantic views based on metapaths, different views carry various semantic information. In real-world HGs, the information contained in each semantic view may be complementary to each other. For example, in an academic network such as DBLP or ACM, we have the four types of nodes: authors (A), papers (P), conferences (C), and topics (T). Considering two metapaths APTPA and APCPA for authors, the two metapaths represent two distinctive aspects of indirect connection between two authors via conferences or topics, respectively. In this case, the two semantic views can discover authors cooperation relationship from two independent aspects (via related conferences and topics). However, it is also plausible that different semantic views share common information. For example, in the above academic network, we consider another metapath APA, denoting two authors share authorship for the same paper. Since authors in the APA metapath are also related to each other in relevant conferences or topics, the semantic information contained in APA and APCPA or APTPC is overlapped.

We further quantitatively demonstrate this observation by analyzing the mutual information (MI) of node embeddings in each semantic view, where the node embeddings are produced by a widely-used supervised model HAN [3]. MI of every two view pairs is estimated using InfoNCE [17] and we plot the node-averaged MI for each semantic view in Figure 1. It is observed that the MI between APCPA and APTPA is the least and the MI between APA and APCPA is much higher, verifying that not all semantic views are complementary to each other, and some of them contain redundant information.

When we construct multiple semantic views by metapaths, it motivates us to *adaptively* encode information from each semantic view. Therefore, we propose to leverage a contrastive aggregation objective for HGs, where we combine information from every semantic views to get an aggregated representation. Following that, we enforce neither cross-view consistency nor disparity, but only retain to discriminate the node embedding per semantic view $\boldsymbol{h}_i^p$ with the aggregated representation $\boldsymbol{h}_j$ and intra-view embeddings $\boldsymbol{h}_j^p$ for other nodes. In this way, the knowledge from each semantic view is encouraged to be distilled to the aggregated representation, which facilitates downstream tasks as a result.



(a) A toy example

(b) Distribution

Fig. 2: (a) A toy example of an academic network. Heterogeneous GNNs produce similar embeddings for nodes sharing the same label in its ego network by aggregating semantic-specific neighborhood information. (b) A histogram of negatives and their semantic similarity scores with an anchor node. With the similarity to the anchor node increasing, there are more positive samples (false negatives), leading to wrong selection of hard negatives.

B. Necessity of Structure-Aware Hard Negative Mining

In CL, for any anchor node, positive and negative samples make identical contribution to the InfoNCE objective. However, previous work [14, 18–20] highlights that hard negative samples, which are more semantically similar to each other, tend to be more helpful for learning the contrastive objective. Therefore, we propose to investigate the relative difficulty of different negative samples in HG scenarios and upweight hard negatives to further boost performance of CL.

In visual CL studies, the hardness of one image is defined to be its semantic similarity to the anchor sample, e.g., inner product of two normalized vectors in the embedding space. For graph data, due to the neighborhood aggregation scheme, GNN produces similar embeddings within ego networks. For the nodes sharing the same class with the anchor node, their embeddings are similar to the anchor, leading to selection of false negatives, as shown in Figure 2(a). Therefore, we argue that the pairwise relationship of node embeddings is insufficient to measure the semantic hardness of each node.

To empirically demonstrate this, we conduct an oracle-based analysis on the DBLP network. Specifically, we plot the relationship between negatives and their similarity scores with one arbitrary anchor node. As shown in Figure 2(b), with the similarity of negative node to the anchor (the hardness) increasing, there are more *positive* samples (*false* negatives). Therefore, measuring semantic hardness simply by embedding similarities results in *hard but false* negatives being selected, which inevitably impairs the performance. Furthermore, at the beginning of training, node embeddings are suffered from poor quality, which may be another obstacle of selecting hard negative samples. The above deficiency motivates us to discover hard negatives for graph-structured data from structural aspects, which mines truly hard negative regardless the training progress.

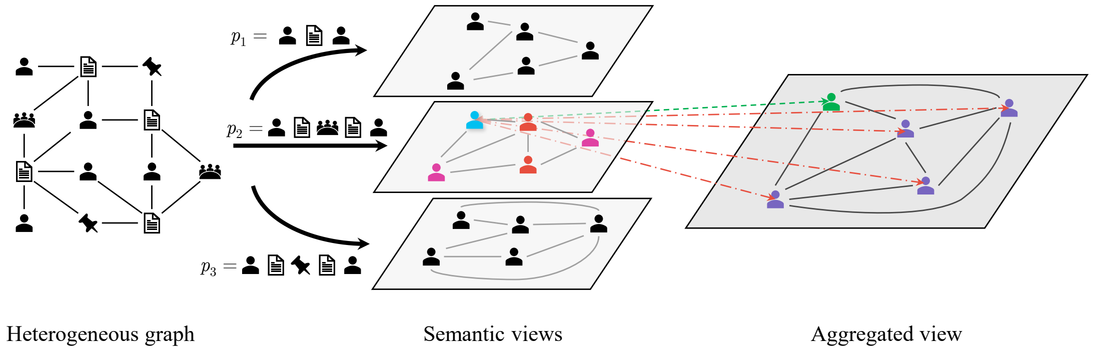


Fig. 3: Illustrating the proposed method. We construct semantic views and learn representations with heterogeneous GNNs (§II-B). Then, we train the model with a multiview contrastive objective (§IV-A). Take $\boldsymbol{h}_1^{p_2}$ as an anchor for example. Its positive sample is the aggregated representation $\boldsymbol{h}_1$; intra-view negatives $\boldsymbol{h}_2^{p_2}$ to $\boldsymbol{h}_5^{p_2}$ and inter-view negatives $\boldsymbol{h}_2$ to $\boldsymbol{h}_5$ constitute its negatives. Structurally hard negatives discovered by our algorithm are highlighted in pink (§IV-B).

IV. The Proposed Method: HORACE

In the following section, we present the proposed HORACE in detail. There are three major components in the proposed HORACE framework: (a) a heterogeneous graph encoder, which embeds each node under each semantic view into low-dimensional vectors and aggregates these semantic-specific embeddings into a final representation, (b) a multiview contrastive aggregation objective that learns node representations in a self-supervised manner, and (c) structure-aware hard negative mining, which discovers and reweights structurally hard samples.

A. Heterogeneous Graph Contrastive Learning via Multiview Contrastive Aggregation

Existing graph CL follows a multiview framework [21–24], which maximizes the agreement among node representations under different views of the original graph and thus enables the encoder to learn informative representations in a self-supervised manner. Following existing heterogeneous GNN approaches, we generate multiple semantic views according to metapaths and learn node representations. Then, since multiple views are involved, the aggregated representations could also be regarded as a view of the original graph.

To comprehensively learn semantics among different views, we propose a novel multiview contrastive aggregation objective, which aims to *maximize the agreement between node representations under a specific semantic view and the aggregated representations*. The contrastive aggregation objective can be mathematically expressed as

$$\max \frac{1}{|\mathcal{V}|} \sum_{v_i \in \mathcal{V}} \left[\frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \frac{1}{2} \left(I(\boldsymbol{h}_i^p; \boldsymbol{h}_i) + I(\boldsymbol{h}_i; \boldsymbol{h}_i^p) \right) \right], \quad (6)$$

where $\boldsymbol{h}_i^p$ is a semantic-specific embedding for node v_i under metapath p and $\boldsymbol{h}_i$ is the aggregated embedding for node v_i that collects information of all its semantic relations.

Following previous work [17, 25], to estimate the mutual information $I(\boldsymbol{h}_i^p; \boldsymbol{h}_i)$ in Eq. (6), we empirically choose the InfoNCE estimator. Specifically, for node representation $\boldsymbol{h}_i^p$ in one specific semantic view, we construct its positive sample as the aggregated representation, while embeddings of all other nodes in the semantic and the aggregated embeddings are considered as negative samples. The contrastive loss can be expressed by

$$\ell(\boldsymbol{h}_i^p, \boldsymbol{h}_i) = -\log \frac{e^{\theta(\boldsymbol{h}_i^p, \boldsymbol{h}_i)/\tau}}{e^{\theta(\boldsymbol{h}_i^p, \boldsymbol{h}_i)/\tau} + \sum_{j \neq i} \left(e^{\theta(\boldsymbol{h}_i^p, \boldsymbol{h}_j)/\tau} + e^{\theta(\boldsymbol{h}_i^p, \boldsymbol{h}_j^p)/\tau} \right)}, \quad (7)$$

where $\tau \in \mathbb{R}$ is a temperature parameter. We define the critic function $\theta(\cdot, \cdot)$ by

$$\theta(\boldsymbol{h}_i, \boldsymbol{h}_j) = \frac{g(\boldsymbol{h}_i)^\top g(\boldsymbol{h}_j)}{\|g(\boldsymbol{h}_i)\| \|g(\boldsymbol{h}_i)\|},$$

where $g(\cdot)$ is parameterized by a non-linear multilayer perceptron to enhance the expressive power [8].

B. Structure-Aware Hard Negative Mining

Previous studies [14–16] demonstrate that CL benefits from hard negative samples, i.e. samples close to the anchor node such that cannot be distinguished easily. In the context of HGs, we observe that semantic-level node representations are not sufficient to calculate the hardness of each negative pair. Therefore, in this work, to effectively measure hardness of each sample with respect to the anchor, we propose to explore the hardness of negative samples in terms of their structural similarities. The proposed structure-aware hard negative mining scheme is illustrated in Figure 4.

We first introduce a structure-aware metric $s(i, j, p)$ representing distance measure of a negative node v_i to the anchor node v_j given a semantic view p, which can be regarded as the hardness of the negative node v_i. Note that in order to empower the model with inductive capabilities, we prefer a local measure to a global one. In this paper, we propose two model variants HORACE-PPR and HORACE-PE, which use Laplacian positional embeddings and personalized PageRank scores for structure-aware hard negative mining respectively.

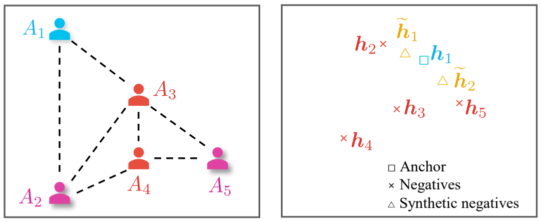


Fig. 4: The proposed structure-aware hard negative mining scheme, which discovers structurally hard negatives in each view and further synthesizes additional harder negative samples.

- The Personalized PageRank (PPR) score [26, 27] of node v is defined as the stationary distribution of a random walk starting from and returning to node v at a probability of c at each step. Formally, the PPR vector of node v under semantic view p satisfies the following equation
$$\boldsymbol{s}_v^p = (1-c)\boldsymbol{A}^p \boldsymbol{s}_v^p + c\boldsymbol{I}\boldsymbol{p}_v, \tag{8}$$
where c is the returning probability and $\boldsymbol{p}_v$ is the preference vector with $(\boldsymbol{p}_v)_i = 1$ when $i = v$ and all other entries set to 0. $\boldsymbol{A}^p$ denotes the adjacency matrix generated by metapath p. The structural similarity between node v and k can be represented by the PPR score of node k with respect to node v, i.e. $(\boldsymbol{s}_v^p)_k$.
- The Laplacian positional embedding of one node is defined to be its k smallest non-trivial eigenvectors [28]. We simply define the structure similarity as the inner product between $\boldsymbol{s}_i^p$ and $\boldsymbol{s}_j^p$.

After that, we perform hard negative mining by giving larger weights to harder negative samples. Specifically, we sort negatives according to the hardness metric and pick the top-T negatives to form a candidate list for semantic view p. Then, we synthesize $M \ll |\mathcal{V}|$ samples by creating a convex linear combination of them. The generated sample $\widetilde{\boldsymbol{h}}_m^p$ is mathematically expressed as

$$\widetilde{\boldsymbol{h}}_m^p = \alpha_m \boldsymbol{h}_i^p + (1-\alpha_m)\boldsymbol{h}_j^p, \tag{9}$$

where $\boldsymbol{h}_i^p, \boldsymbol{h}_j^p \in \mathcal{B}^p$ are randomly picked from the memory bank, $\alpha_m \sim \operatorname{Beta}(\alpha,\alpha)$, and α is a hyperparameter, fixed to 1 in our experiments. These interpolated samples will be added into negative bank when estimating mutual information $I(\boldsymbol{h}_i^p; \boldsymbol{h}_i)$, as given in sequel

$$\mathcal{L}(\boldsymbol{h}_i^p, \boldsymbol{h}_i) = -\log \frac{e^{\theta(\boldsymbol{h}_i^p, \boldsymbol{h}_i)/\tau}}{e^{\theta(\boldsymbol{h}_i^p, \boldsymbol{h}_i)/\tau} + \sum\limits_{\boldsymbol{h} \in \mathcal{B}^p} e^{\theta(\boldsymbol{h}_i^p, \boldsymbol{h})/\tau}}, \tag{10}$$

where the negative bank

$$\mathcal{B}^p = \{\boldsymbol{h}_j^p\}_{j \neq i} \cup \{\boldsymbol{h}_j\}_{j \neq i} \cup \{\widetilde{\boldsymbol{h}}_m^p\}_{m=1}^M \tag{11}$$

consists of all inter-view and intra-view negatives as well as synthesized hard negatives. The contrastive objective $\ell(\boldsymbol{h}_i; \boldsymbol{h}_i^p)$ for the aggregated node representation $\boldsymbol{h}_i$ can be defined similarly as Eq. (10). The final objective is an average of the losses from all contrastive pairs, formally given by

$$\mathcal{J} = \frac{1}{|\mathcal{V}|} \sum_{v_i \in \mathcal{V}} \left[\frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \frac{1}{2} \left(\mathcal{L}(\boldsymbol{h}_i^p; \boldsymbol{h}_i) + \mathcal{L}(\boldsymbol{h}_i; \boldsymbol{h}_i^p) \right) \right]. \tag{12}$$

We use stochastic gradient descent algorithms to update all model parameters. Finally, we summarize the training procedure of the proposed HORACE in Algorithm 1.

Algorithm 1: The HORACE framework

```
Construct multiple semantic views corresponds to
  metapath $p \in \mathcal{P}$
for epoch $\leftarrow 1, 2, \cdots$ do
    /* Heterogeneous graph encoding */
    Obtain node embeddings of each semantic view $\boldsymbol{H}^p$
      according to Eq. (1)
    Obtain aggregated embeddings $\boldsymbol{H}$ according to Eq.
      (3)
    /* Structure-aware hard negative mining */
    Compute hardness score $S = s(i,j,p) + c(i,j,p)$
      for each negative-anchor pair
    Sort $S$ in ascending order
    Pick $T$ negative nodes with the highest $S$ in each
      semantic view
    Synthesis $M$ hard negative samples via Eq. (9)
    Update the negative bank $\mathcal{B}$ according to Eq. (V-C1)
    /* Model training */
    Compute the contrastive objective given in Eq. (6)
    Update parameters by applying stochastic gradient
      descent to minimize $\mathcal{J}$ as in Eq. (12)
```

C. Complexity Analysis

Most computational burden of the HORACE framework lies in the contrastive objective, which involves computing $(|\mathcal{V}|^2|\mathcal{P}|)$ node embedding pairs. For structure-aware hard negative mining, the synthesized samples incur an additional computational cost of $O(M|\mathcal{V}||\mathcal{P}|)$, which is equivalent to increasing the memory size by $M \ll |\mathcal{V}|$. The construction of the candidates list of hard negatives only depends on graph structures of each semantic view, and thus it can be regarded as a preprocessing process.

D. Discussions with Existing Work

The proposed multiview contrastive aggregation objective Eq. (6) conceptually relates to contrastive knowledge distillation [29], where several teacher models (semantic views) and one student model (the aggregated view) are employed. By forcing the embeddings between several teachers and a student to be the same, these aggregated embeddings adaptively collect information of all semantic relations.

Moreover, the proposed structure-aware hard negative mining scheme generally resembles many studies in domains of metric learning [16, 18] and visual contrastive learning [15, 20, 30, 31].

TABLE I: Statistics and sources of the public datasets used in experiments.

Dataset	Node	Relations	Metapaths
DBLP[1]	Paper (14,328) Author (4,057) Conference (20) Term (8,789)	P–A (19,645) P–C (14,328) P–T (88,420)	APA APCPA APTPA
ACM[2]	Paper (3,025) Author (5,835) Subject (56)	P–A (9,744) P–S (3,025)	PAP PSP
IMDb[3]	Movie (4,780) Actor (5,841) Director (2,269)	M–A (14,340) M–D (4,780)	MAM MDM

[1] http://ews.uiuc.edu/~jinggao3/doc/BGCM.zip

[2] https://github.com/Jhy1993/HAN/blob/master/data/acm/ACM.mat

[3] https://github.com/Jhy1993/HAN/blob/master/data/IMDb/movie_metadata.csv

Nevertheless, none of these methods can be applied to graph-structured data, as the hardness score defined simply by inner product of node representations is not sufficient to distinguish hard negative nodes in graphs and it even results in amplifying false negatives.

V. Experiments

We evaluate the effectiveness of our proposed HORACE in this section. The purpose of empirical studies is to answer the following questions.

- **RQ1**. How does our proposed HORACE outperform other representative baseline algorithms?
- **RQ2**. How does the proposed structure-aware hard negative mining scheme affect the performance of HORACE?
- **RQ3**. How sensitive are key hyperparameters in the proposed HORACE model?

A. Experimental Configurations

1) Datasets: To achieve a comprehensive comparison, we use three widely-used heterogeneous datasets from different domains: DBLP, ACM, and IMDb, where DBLP and ACM are two academic networks, and IMDb is a movie network. The statistics of three used datasets is summarized in Table I.

- **DBLP** is a subset of an academic network extracted from DBLP, consisting of four kinds of nodes: authors, papers, conferences, and topics. The authors are selected from four domains: database, data mining, machine learning, and information retrieval. Each author is labeled with their research area according to the conferences they submitted, and is associated with bag-of-word features which represent keywords.
- **ACM** is an academic network extracted from papers published in KDD, SIGMOD, SIGCOMM, MobiCOMM, and VLDB. We construct a heterogeneous graph with nodes of three types: papers, authors, and subjects. Papers with bag-of-words of features are classified into three themes according to their corresponding research topic.
- **IMDb** is a subset of the movie network IMDb, where nodes represent movies, actors, or directors. We categorize movies into three classes according to their genre. Each movie node is associated with a bag-of-words feature representing plots.

2) Baselines: We compare the proposed HORACE against a comprehensive set of baselines, including both representative traditional and deep graph representation learning methods.

- **DeepWalk** [32] is a widely-used homogeneous model that generates several sequences by random walk. It is trained using the skip-gram objective [33].
- **ESim** [34] captures node semantics from sampled metapath instance with a preset weight. In our experiments, we simply treat all metapaths equally.
- **metapath2vec** [35] performs metapath-based random walks and learns node representations using the skip-gram model as DeepWalk. Since metapath2vec only utilizes one metapath, we experiment with all metapaths and report the best preformance.
- **HERec** [1] converts the heterogeneous graph into metapath-based graphs and utilizes the skip-gram model to embed the heterogeneous graph. Similar to metapath2vec, we test all metapaths and report the best performance.
- **GCN** [4] is a deep-learning-based semi-supervised baseline for homogeneous graphs, which works by aggregating information from neighborhoods.
- **GAT** [36] is also a semi-supervised baseline designed for homogeneous graphs. It further leverages the self-attention mechanism to model anisotropic neighborhood information.
- **HAN** [3] is a semi-supervised baseline for heterogeneous graphs, which proposes node- and semantic-level attention for learning node representations. We also include the unsupervised version of HAN (denoted by **HAN-U**) trained with link prediction loss, for further comparison with our proposed contrastive learning objective.
- **DGI** [5] is a deep contrastive learning model for homogeneous graphs, which maximizes the agreement of node representations and a global summary vector.
- **GRACE** [21] is the state-of-the-art contrastive learning model for homogeneous graphs. It uses a node-level contrastive objective by generating two graph views and maximizing the agreement between them.

Among these baselines, DeepWalk, DGI, GRACE, GCN, and GAT are designed for homogeneous graphs, and the others are for heterogeneous graphs. Following HAN [3], for DeepWalk, we simply discard node and edge types, and treat the heterogeneous graph as a homogeneous graph; for DGI, GRACE, GCN, and GAT, we generate homogeneous graphs according to all metapaths, and report the best performance.

3) Implementation details: The proposed model is implemented using PyTorch [37], DGL [38], and PyTorch Geometric [39]. We use Adam optimizer [40] with learning rate set to 0.01, 0.0005, and 0.001 for ACM, IMDb and DBLP respectively and ℓ_2 regularization set to 10^{-5}. The model is trained for at most 3,000 epochs and is early-stopped if the training loss does not improve for 100 consecutive epochs. The dropout rate [41] is

TABLE II: Performance comparison on three datasets. Node classification performance is in terms of Macro-F1 (Ma-F1) and Micro-F1 (Mi-F1). Node clustering performance is in terms of Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). Available training data is shown in the second column, where $\boldsymbol{A}$ denotes adjacency matrices according to metapaths, $\boldsymbol{X}$ denotes node features, and $\boldsymbol{Y}$ denotes labels. The highest performance of unsupervised and supervised models is boldfaced and underlined, respectively.

Method	Training Data	Node Classification						Node Clustering					
		ACM		IMDb		DBLP		ACM		IMDb		DBLP	
		Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	NMI	ARI	NMI	ARI	NMI	ARI
DeepWalk	$\boldsymbol{A}$	76.92	77.25	46.38	40.72	79.37	77.43	41.61	35.10	1.45	2.15	76.53	81.35
ESim	$\boldsymbol{A}$	76.89	77.32	35.28	32.10	92.73	91.64	39.14	34.32	0.55	0.10	66.32	68.31
metapath2vec	$\boldsymbol{A}$	65.00	65.09	45.65	41.16	91.53	90.76	21.22	21.00	1.20	1.70	74.30	78.50
HERec	$\boldsymbol{A}$	66.03	66.17	45.81	41.65	92.69	91.78	40.70	37.13	1.20	1.65	**76.73**	78.50
HAN-U	$\boldsymbol{A}, \boldsymbol{X}$	82.63	81.89	43.98	40.87	90.47	89.65	39.84	32.98	3.92	4.10	74.17	79.98
DGI	$\boldsymbol{A}, \boldsymbol{X}$	89.15	89.09	48.86	45.38	91.30	90.69	58.13	57.18	8.31	11.25	60.62	60.42
GRACE	$\boldsymbol{A}, \boldsymbol{X}$	88.72	88.72	46.64	42.41	90.88	89.76	53.38	54.39	7.52	9.16	62.06	64.13
HORACE-PE	$\boldsymbol{A}, \boldsymbol{X}$	**90.76**	**90.72**	**58.98**	**54.48**	**92.81**	**92.33**	67.93	72.65	**15.09**	**17.23**	76.60	**81.58**
HORACE-PPR	$\boldsymbol{A}, \boldsymbol{X}$	90.75	90.70	58.96	54.47	92.78	92.30	**68.10**	**73.15**	15.03	17.09	76.52	81.49
GCN	$\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{Y}$	86.77	86.81	49.78	45.73	91.71	90.79	51.40	53.01	5.45	4.40	75.01	80.49
GAT	$\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{Y}$	86.01	86.23	55.28	49.44	91.96	90.97	57.29	60.43	8.45	7.46	71.50	77.26
HAN	$\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{Y}$	$\underline{89.22}$	$\underline{89.40}$	$\underline{54.17}$	$\underline{49.78}$	$\underline{92.05}$	$\underline{91.17}$	$\underline{61.56}$	$\underline{64.39}$	$\underline{10.31}$	$\underline{9.51}$	$\underline{79.12}$	$\underline{84.76}$

set to 0.2 on all datasets. We use 8 attention heads and the embedding size is 64 for both HORACE and baselines for fair comparison. Furthermore, we set the temperature parameter τ to 0.9 in the contrastive objective. The number of synthesize samples M is set to 200. All parameters are initialized with Glorot initialization [42].

B. Performance Comparison (RQ1)

For comprehensive evaluation, we follow HAN [3] and perform experiments on two tasks: node classification and node clustering.

1) Evaluation protocols: For node classification, we run a k-NN classifier with $k = 5$ on the learned node embeddings. We report performance in terms of Micro-F1 and Macro-F1 for evaluation of node classification. For dataset split, we randomly pick 20% nodes in each dataset for training and the remaining 80% for test. Results from 10 different random splits are averaged for the final report.

Regarding node clustering, we run k-Means algorithm on the learned node embeddings with k set to the number of ground-truth classes. NMI and ARI of the obtained clusters with respect to ground-truth classes are the evaluation metrics for clustering. Since the results of k-Means are highly sensitive to initialization, we run the clustering algorithm for 10 times and report the averaged performance.

2) Performance and analysis: Experiment results are presented in Table II. Overall, our proposed HORACE achieves the best unsupervised performance on almost all datasets on both node classification and clustering tasks. It is worth mentioning that our HORACE is competitive to and even better than several representative supervised counterparts.

Regarding two model variants HORACE-PE and HORACE-PPR, their performance different is negligible, which demonstrate that both Laplacian positional embedding and Personalized PageRank score could be used to calculate local structural similarities and are suitable for structure-aware hard negative mining.

Compared with traditional approaches based on random walks and matrix decomposition, our proposed GNN-based HORACE outperforms them by large margins. Particularly, HORACE improves metapath2vec and HERec by over 25% on ACM, which demonstrates the superiority of GNN that can leverage rich node attributes to learn high quality node representations for heterogeneous graphs.

For deep unsupervised learning methods, our HORACE achieves promising improvements as well. For the unsupervised version HAN-U that is trained with a simple reconstruction loss, its performance is even inferior to HERec on IMDb and DBLP despite its utilization of node attributes. This indicates that the reconstruction loss is insufficient to fully exploit the structural and semantic information for node-centric tasks such as node classification and clustering. Compared to DGI and GRACE, two homogeneous contrastive learning methods, HORACE accomplishes excelled performance on all datasets and evaluation tasks, especially on ACM and IMDb dataset, where large improvements on both tasks are achieved. This validates the effectiveness of our proposed view-to-aggregation contrastive objective and structure-aware hard negative mining strategy.

Furthermore, experiments show that HORACE even outperforms its supervised baselines on ACM and IMDb datasets. It remarkably improves HAN by over 4% in terms of node classification Micro-F1 score on IMDb. This outstanding performance of HORACE certifies the superiority of our proposed HGCL framework such that it can distill useful information from each semantic view.

TABLE III: Effectiveness of the structure-aware hard negative mining module.

Method	Node Classification						Node Clustering					
	ACM		IMDb		DBLP		ACM		IMDb		DBLP	
	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	NMI	ARI	NMI	ARI	NMI	ARI
HORACE–	88.62	88.43	57.94	52.97	92.42	91.85	58.08	61.80	14.15	15.98	76.23	81.43
HORACE-Sem	90.24	90.18	58.95	52.38	92.73	92.21	51.63	48.85	15.17	17.25	76.22	81.15
HORACE-PE	**91.40**	**91.45**	**58.96**	**53.73**	**92.77**	**92.28**	**66.57**	**72.30**	**15.36**	**17.30**	**76.59**	**81.56**

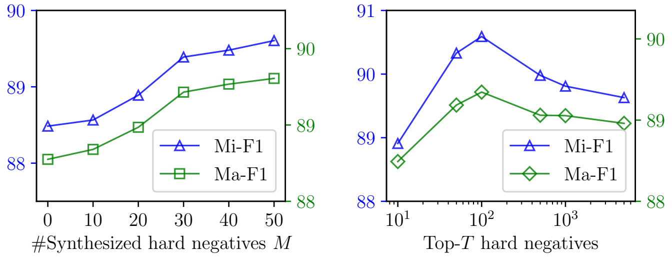
C. Close Inspections on Structure-Aware Hard Negative Mining Module (RQ2)

1) Effectiveness of the module: We modify the negative bank in our contrastive objective to study the impact of structure-aware hard negative mining component. HORACE– denotes the model with synthesized harder samples $\{\widetilde{\boldsymbol{h}}_m^p\}_{m=1}^M$ removed, where the negative bank $\mathcal{B}^p = \{\boldsymbol{h}_j^p\}_{j \neq i} \cup \{\boldsymbol{h}_j\}_{j \neq i}$ consists of only inter-view and intra-view negatives. We also construct a model variant HORACE-Sem, that discovers and synthesizes semantic negative samples using inner product of node embeddings.

The results are presented in Table III. It is observed that HORACE improves all two model variants consistently on three datasets for both node classification and clustering tasks. Especially for node clustering task on ACM, the gain reaches up to 15%. This verifies the effectiveness of our synthesizing hard negative sample strategy: giving larger weights to harder negative samples with the delicately designed synthesis term. Secondly, we see that the performance of HORACE-Sem slightly improves the base model on several times, which demonstrates the importance of hard negative mining in effective CL. However, its performance is still inferior to that of our proposed model. The outstanding performance of HORACE compared to the model variant HORACE-Sem further justifies the superiority of our proposed structure-aware hard negative mining which exploits the abundant structural information of HGs.

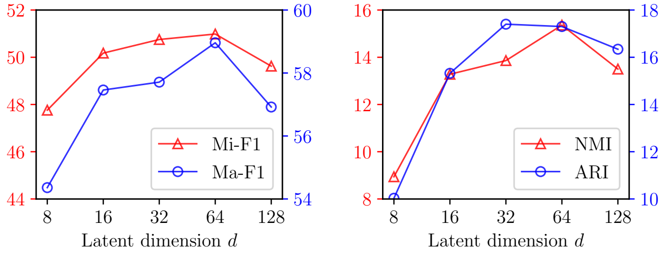
2) The impact of two key parameters in the module: We study how the two key parameters in the hard negative mining module affect the performance of HORACE: the number of synthesized hard negatives M and the threshold T in selecting top-T candidate hard negatives. We perform node classification on the ACM dataset under different parameter settings by only varying one specific parameter and keeping all other parameters the same. The results are summarized in Figure 5.

As is shown in Figure 5a, the performance of HORACE improves as the number of synthesized negatives M increases. This indicates that the learning of HORACE benefits from the synthesized hard negatives. For the parameter T, as presented in Figure 5b, the model performance first rises with a larger T, but soon the performance levels off and decreases as T increases further. We suspect that this is because a larger T will result in the selection of less hard negatives, reducing the benefits brought by our proposed hard negative sampling strategy.



(a) Synthesized hard negatives (b) Candidate hard negative samples

Fig. 5: Node classification performance with varied numbers of synthesized hard negatives and candidate hard negative samples T on the ACM dataset.



(a) Node Classification (b) Node Clustering

Fig. 6: Model performance with varied latent dimensions.

D. Sensitivity Analysis (RQ3)

In this section, we perform sensitivity study on one key hyperparameter in our proposed HORACE model, namely the dimension of hidden representation d. Note that all other parameters described previously remain the same while we are varying a specific parameter, to show the model stability under the perturbation of each hyperparameter. Two downstream tasks, node classification and node classification, are included using the corresponding evaluation metrics and the results are on the IMDb dataset.

We show the influence of varied node latent dimensions d on HORACE in Figure 6. It is observed that at initial stages, the performance of HORACE on both two tasks improves noticeably as the latent dimension increases. This is because that the model can encode richer information with larger dimension size, which facilitate the performance on various downstream tasks. However, as d continues to grow, the improvement level of Mi-F1 and ARI gets scaled down, and finally the performance of four different metrics on two tasks begin to descend. The reason may be that with more latent dimensions, the model gets sized-up and harder to train, which possibly leads to under-fitting with the same amount of training samples. Therefore, we need to choose a moderate and appropriate dimension d for balancing the expressiveness and size, as well as the efficiency of our model.

VI. Related Work

This section reviews previous related work on heterogeneous graph embedding methods. Following that, we discuss recent work on graph contrastive learning.

A. Heterogeneous Graph Embedding

The purpose of Heterogeneous Graph Embedding (HGE) is to project nodes in a heterogeneous graph into a low-dimensional embedding space that preserves structural and semantic information. Most work of HGE could be grouped into two lines of development: proximity-preserving methods and deep learning approaches. Readers of interest may refer to [43] for a comprehensive survey on heterogeneous network representation learning.

a) Proximity-preserving methods: Inspired by network embedding methods for homogeneous graphs, traditional HGE methods roughly fall into two lines: random-walk-based approaches and methods based on preserving first-/second-order proximity. On the one hand, originated from random-walk-based methods for homogeneous graphs [32, 44], metapath2vec [35] models node context via metapath-based random walks and learns node embeddings using the skip-gram model [33]. Similarly, HERec [1] transforms a heterogeneous graph into a homogeneous one through metapath-based neighborhood and learns representations using DeepWalk-like strategies. HIN2Vec [45] further proposes a multitask learning objective to learn representations for nodes and metapaths simultaneously. On the other hand, the pioneering proximity-preserving method PTE [46] extends LINE [46] to heterogeneous text graphs. HEER [13] further improves PTE by considering type closeness via edge representations. These aforementioned traditional approaches could be regarded as shallow embedding and thus have difficulty in leveraging rich node attributes, due to the fact that they are essentially factorizing a certain proximity matrix [47].

b) Deep learning approaches: Recent years have witnessed the surge of Graph Neural Networks (GNN) [4, 36], which proposes to learn representations by aggregating features from node neighborhoods. There has been many attempts adopting GNN into heterogeneous graphs. To name a few, R-GCN [48] introduces multiple graph convolutional layers, each corresponds to one edge type. GTN [49] firstly generates all possible connections via graph transformer layers and performs graph convolution on the new graph afterwards. Following GAT [36], HAN [3] introduces self-attention mechanisms [50] to aggregate features from metapath-based neighborhoods and weigh different metapaths. Similarly, HetGNN [51] adopts node-type-based neighborhood aggregation, where the neighborhood is sampled using random walk with restart. Moreover, MAGNN [6] further proposes to aggregate intermediate node features along each metapath. When performing neighborhood aggregation, HGT [52] implicitly learns metapaths by modeling heterogeneous attention over each edge.

B. Graph Contrastive Learning

Recently, considerable attention has grown up around the theme of graph contrastive learning, which marries the power of GNN and unsupervised learning. We refer readers to [53, 54] for a comprehensive survey.

The very first work DGI [5] proposes to maximizes mutual information (ML) between node embeddings and a global summary embedding. To be specific, DGI constructs negative graphs by random shuffling node attributes. Then, it requires an injective readout function to produce a graph-level embedding. Mirroring DGI, HDGI [55] adopts CL into heterogeneous graphs. However, the injective property is hard to fulfill in practice and thus these methods may cause information loss due to non-injectivity. Follow-up work GRACE [21] and GraphCL [23] eschew the need of an injective readout function and propose a node-level contrastive framework. Following their work, GCA [22] further proposes several augmentation schemes that are adaptive to graph structures and attributes. However, these methods consider all negative samples to be equal, leading to suboptimal performance. Our work, on the contrary, explicitly conducts hard negative mining, which is proved to be a useful technique to boost performance in learning representations of visual data [15, 20, 30, 31, 56]. Moreover, we argue that inner product of node embeddings is inefficient to encode similarity between nodes. In our work, we propose to define hardness of examples via structural similarity, which yields harder negative samples in the context of heterogeneous graphs.

VII. Conclusion

This paper has developed a novel heterogeneous graph contrastive learning framework. To alleviate the label scarcity problem, we leverage contrastive learning techniques that enables self-supervised training for HGs. Specifically, we propose a novel multiview contrastive aggregation objective that encodes information adaptively from each semantic view. Furthermore, we propose a novel hard negative mining scheme to improve the embedding quality, considering the complex structure of heterogeneous graphs and smoothing nature of heterogeneous GNNs. The proposed structure-aware negative mining scheme discovers and reweights structurally hard negatives so that they contribute more to contrastive learning. Extensive experiments have been conducted on three real-world heterogeneous datasets. The experimental results show that our proposed method not only consistently outperforms representative unsupervised baseline methods, but also achieves on par performance with supervised counterparts, and is even superior to several of them.

References

[1] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, “Heterogeneous Information Network Embedding for Recommendation,” *TKDE*, vol. 31, no. 2, pp. 357–370, 2019.

[2] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han, “Co-author Relationship Prediction in Heterogeneous Bibliographic Networks,” in *ASONAM*, 2011, pp. 121–128.

[3] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, “Heterogeneous Graph Attention Network,” in *WWW*, 2019, pp. 2022–2032.

[4] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” in *ICLR*, 2017.

[5] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, “Deep Graph Infomax,” in *ICLR*, 2019.

[6] X. Fu, J. Zhang, Z. Meng, and I. King, “MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding,” in *WWW*, 2020, pp. 2331–2341.

[7] T. Xiao, X. Wang, A. A. Efros, and T. Darrell, “What Should Not Be Contrastive in Contrastive Learning,” in *ICLR*, 2021.

[8] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, “A Simple Framework for Contrastive Learning of Visual Representations,” in *ICML*, 2020, pp. 1597–1607.

[9] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, “Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning,” in *NeurIPS*, 2020.

[10] X. Chen and K. He, “Exploring Simple Siamese Representation Learning,” *arXiv.org*, Nov. 2020.

[11] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, “What Makes for Good Views for Contrastive Learning,” in *NeurIPS*, 2020.

[12] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised Learning of Visual Features by Contrasting Cluster Assignments,” in *NeurIPS*, 2020.

[13] Y. Shi, Q. Zhu, F. Guo, C. Zhang, and J. Han, “Easing Embedding Learning by Comprehensive Transcription of Heterogeneous Information Networks,” in *KDD*, 2018, pp. 2190–2199.

[14] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering,” in *CVPR*, 2015, pp. 815–823.

[15] T. T. Cai, J. Frankle, D. J. Schwab, and A. S. Morcos, “Are All Negatives Created Equal in Contrastive Instance Discrimination?” *arXiv.org*, Oct. 2020.

[16] H. Xuan, A. Stylianou, X. Liu, and R. Pless, “Hard Negative Examples are Hard, but Useful,” in *ECCV*, 2020, pp. 126–142.

[17] A. van den Oord, Y. Li, and O. Vinyals, “Representation Learning with Contrastive Predictive Coding,” *arXiv.org*, 2018.

[18] B. Harwood, B. G. V. Kumar, G. Carneiro, I. D. Reid, and T. Drummond, “Smart Mining for Deep Metric Learning,” in *ICCV*, 2017, pp. 2840–2848.

[19] N. Saunshi, O. Plevrakis, S. Arora, M. Khodak, and H. Khandeparkar, “A Theoretical Analysis of Contrastive Unsupervised Representation Learning,” in *ICML*, 2019, pp. 5628–5637.

[20] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, “Hard Negative Mixing for Contrastive Learning,” in *NeurIPS*, 2020.

[21] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, “Deep Graph Contrastive Representation Learning,” in *GRL+@ICML*, Jun. 2020.

[22] ——, “Graph Contrastive Learning with Adaptive Augmentation,” in *WWW*, 2021, pp. 2069–2080.

[23] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, “Graph Contrastive Learning with Augmentations,” in *NeurIPS*, 2020.

[24] K. Hassani and A. H. Khasahmadi, “Contrastive Multi-View Representation Learning on Graphs,” in *ICML*, 2020, pp. 4116–4126.

[25] O. J. Hénaff, A. Srinivas, J. De Fauw, A. Razavi, C. Doersch, S. M. A. Eslami, and A. van den Oord, “Data-Efficient Image Recognition with Contrastive Predictive Coding,” in *ICML*, 2020, pp. 4182–4192.

[26] G. Jeh and J. Widom, “Scaling Personalized Web Search,” in *WWW*, 2003.

[27] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web,” Tech. Rep., Nov. 1999.

[28] V. P. Dwivedi, C. K. Joshi, T. Laurent, Y. Bengio, and X. Bresson, “Benchmarking Graph Neural Networks,” *arXiv.org*, Mar. 2020.

[29] Y. Tian, D. Krishnan, and P. Isola, “Contrastive Representation Distillation,” in *ICLR*, 2020.

[30] M. Wu, M. Mosse, C. Zhuang, D. Yamins, and N. Goodman, “Conditional Negative Sampling for Contrastive Learning of Visual Representations,” in *ICLR*, 2021.

[31] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, “Contrastive Learning with Hard Negative Samples,” *arXiv.org*, Oct. 2020.

[32] B. Perozzi, R. Al-Rfou, and S. Skiena, “DeepWalk: Online Learning of Social Representations,” in *KDD*, 2014, pp. 701–710.

[33] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” in *ICLR*, 2013.

[34] J. Shang, M. Qu, J. Liu, L. M. Kaplan, J. Han, and J. Peng, “Meta-Path Guided Embedding for Similarity Search in Large-Scale Heterogeneous Information Networks,” *arXiv.org*, Oct. 2016.

[35] Y. Dong, N. V. Chawla, and A. Swami, “metapath2vec: Scalable Representation Learning for Heterogeneous Networks,” in *KDD*, 2017, pp. 135–144.

[36] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph Attention Networks,” in *ICLR*, 2018.

[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *NeurIPS*, 2019, pp. 8024–8035.

[38] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, “Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks,” *arXiv.org*, Sep. 2019.

[39] M. Fey and J. E. Lenssen, “Fast Graph Representation Learning with PyTorch Geometric,” in *RLGM@ICLR*, 2019.

[40] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *ICLR*, 2015.

[41] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks From Overfitting,” *JMLR*, vol. 15, no. 1, pp. 1929–1958, 2014.

[42] X. Glorot and Y. Bengio, “Understanding the Difficulty of Training Deep Feedforward Neural Networks,” in *AISTATICS*, 2010, pp. 249–256.

[43] C. Yang, Y. Xiao, Y. Zhang, Y. Sun, and J. Han, “Heterogeneous Network Representation Learning: A Unified Framework with Survey and Benchmark,” *TKDE*, 2021.

[44] A. Grover and J. Leskovec, “node2vec: Scalable Feature Learning for Networks,” in *KDD*, 2016, pp. 855–864.

[45] T.-Y. Fu, W.-C. Lee, and Z. Lei, “HIN2Vec: Explore Meta-paths in Heterogeneous Information Networks for Representation Learning,” in *CIKM*, 2017, pp. 1797–1806.

[46] J. Tang, M. Qu, and Q. Mei, “PTE: Predictive Text Embedding through Large-scale Heterogeneous Text Networks,” in *KDD*, 2015, pp. 1165–1174.

[47] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, “Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec,” in *WSDM*, 2018, pp. 459–467.

[48] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling Relational Data with Graph Convolutional Networks,” in *ESWC*, 2018, pp. 593–607.

[49] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, “Graph Transformer Networks,” in *NeurIPS*, 2019, pp. 11 960–11 970.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, U. Kaiser, and I. Polosukhin, “Attention is All You Need,” in *NIPS*, 2017, pp. 5998–6008.

[51] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, “Heterogeneous Graph Neural Network,” in *KDD*, 2019, pp. 793–803.

[52] Z. Hu, Y. Dong, K. Wang, and Y. Sun, “Heterogeneous Graph Transformer,” in *WWW*, 2020, pp. 2704–2710.

[53] Y. Liu, S. Pan, M. Jin, C. Zhou, F. Xia, and P. S. Yu, “Graph Self-Supervised Learning: A Survey,” *arXiv.org*, Feb. 2021.

[54] L. Wu, H. Lin, Z. Gao, C. Tan, and S. Z. Li, “Self-supervised on Graphs: Contrastive, Generative, or Predictive,” *arXiv.org*, May 2021.

[55] Y. Ren, B. Liu, C. Huang, P. Dai, L. Bo, and J. Zhang, “Heterogeneous Deep Graph Infomax,” *arXiv.org*, Nov. 2019.

[56] C.-Y. Chuang, J. Robinson, L. Yen-Chen, A. Torralba, and S. Jegelka, “Debiased Contrastive Learning,” in *NeurIPS*, 2020.