
HGOT: Self-supervised Heterogeneous Graph Neural Network with Optimal Transport

Yanbei Liu¹ Chongxu Wang¹ Zhitao Xiao¹ Lei Geng¹ Yanwei Pang² Xiao Wang³

Abstract

Heterogeneous Graph Neural Networks (HGNNs), have demonstrated excellent capabilities in processing heterogeneous information networks. Self-supervised learning on heterogeneous graphs, especially contrastive self-supervised strategy, shows great potential when there are no labels. However, this approach requires the use of carefully designed graph augmentation strategies and the selection of positive and negative samples. Determining the exact level of similarity between sample pairs is non-trivial. To solve this problem, we propose a novel self-supervised Heterogeneous graph neural network with Optimal Transport (HGOT) method which is designed to facilitate self-supervised learning for heterogeneous graphs without graph augmentation strategies. Different from traditional contrastive self-supervised learning, HGOT employs the optimal transport mechanism to relieve the laborious sampling process of positive and negative samples. Specifically, we design an aggregating view (central view) to integrate the semantic information contained in the views represented by different meta-paths (branch views). Then, we introduce an optimal transport plan to identify the transport relationship between the semantics contained in the branch view and the central view. This allows the optimal transport plan between graphs to align with the representations, forcing the encoder to learn node representations that are more similar to the graph space and of higher quality. Extensive experiments on four real-world datasets demonstrate that our proposed HGOT

model can achieve state-of-the-art performance on various downstream tasks. In particular, in the node classification task, HGOT achieves an average of more than 6% improvement in accuracy compared with state-of-the-art methods.

1. Introduction

Heterogeneous graphs are prevalent in numerous real-world contexts, including academic networks (Wang et al. 2019), biomedical networks (Bai et al. 2021), social networks (Cao et al. 2021) and so on. These graphs serve to represent diverse relationships among various types of nodes. In recent years, a range of heterogeneous graph neural networks (HGNNs) has been developed to effectively capture the intricate structures and abundant semantic information inherent in these graphs, demonstrating commendable performance in the domain of heterogeneous graph representation learning (Wang et al. 2020, Dong et al. 2020). To date, HGNNs have found extensive applications across a variety of fields (Tian et al. 2022a, Bansal et al. 2019, Wang et al. 2021a).

The majority of current HGNNs rely on supervised or semi-supervised learning methods (Sun et al. 2025, Ji et al. 2024), necessitating the guidance of labeled data throughout the model’s learning process. However, acquiring node labels from heterogeneous graph data poses significant challenges and incurs high costs in many applications (Hu et al. 2020). Consequently, in the absence of labeled data, self-supervised learning (SSL) emerges as a viable alternative, as it seeks to derive valuable supervisory signals directly from the input data.

In the realm of self-supervised learning applied to heterogeneous graphs, contrastive learning has emerged as a prominent technique. This approach involves carefully designed graph augmentation strategies and the selection of positive and negative samples. The objective is to enhance the similarity between positive samples while concurrently reducing the similarity between negative samples. Some works (Zhao et al. 2024, Zhu et al. 2022) on heterogeneous graph contrastive learning propose different graph augmentation strategies to better capture the underlying high-order semantic structure information. In addition, many methods

¹School of Life Sciences, Tiangong University, Tianjin, China

²School of Electrical and Information Engineering, Tianjin University, Tianjin, China ³School of Software, Beihang University, Beijing, China. Correspondence to: Zhitao Xiao <xiaozhitao@tiangong.edu>, Xiao Wang <xiao.wang@buaa.edu>.

(Wang et al, 2021a, Zhao et al. 2024) retain the traditional contrastive learning and design the method of selecting the required positive and negative sample pairs from different views or different structures, and can more effectively obtain high-level factors. Although the contrastive self-supervised learning strategy enhances the quality of node representations in HGNNs and facilitates their application in real-world scenarios devoid of label supervision (Xie et al. 2024, Liu et al. 2024, Chen et al. 2024), two key challenges persist in the contrastive self-supervised learning for heterogeneous graphs:

- **C1: More effective self-supervised learning strategy for heterogeneous graphs.** On the one hand, existing contrastive self-supervised methods on heterogeneous graph require graph augmentations. The heterogeneous graph data structure is discrete, so even if the graph is slightly perturbed, the properties of the graph may change, which leads to inconsistent labels between the original and augmented graphs. Therefore, it is challenging to design reasonable augmentations that guarantee the label consistency for heterogeneous graphs. On the other hand, the strategy of contrastive learning is to maximize the similarity for positive pairs of the same anchor and minimize the similarity for those of negative pairs. However, the concept of “maximum or minimum similarity” is difficult to measure since it is vague and lacks a clear indication of how much similarity should be maximized (or minimized) for a given pair of positive (or negative) views.
- **C2: More comprehensive heterogeneous semantic information aggregation.** Heterogeneous graphs typically exhibit complex structural variations, e.g., objects of different types, diverse relationships, and varying degrees of connectivity. Integrating information across these structures while maintaining the integrity of the graph’s overall topology (i.e., the interconnections between objects) presents a further difficulty. A simplistic aggregation approach could distort the graph’s structural properties, leading to a loss of context and nuance.

To address these challenges, in this paper, we propose HGOT, a novel self-supervised learning strategy for heterogeneous graphs that does not require graph augmentations and positive or negative samples. We exploits the key concept in optimal transport (OT) theory, which aims to identify the matching relationship between graph and representation. In order to eliminate the diversity of node types, we design the node feature transformation to project all nodes into a unified dimensional feature space. To further capture more comprehensive semantics, we employ the attention mechanism to aggregate different meta-paths

(branch views) to obtain an aggregated view (central view) which can integrate the semantic information from different meta-paths. We then introduce an optimal transport plan to identify the transport relationship between the semantics contained in each branch view and the central view. To accurately capture the matching relationships in the original graph space, the encoders are forced to learn representations that exhibit consistency with the optimal transport plans between the corresponding meta-path views and aggregated view. In summary, the main contributions of this work are as follows:

- To our best knowledge, this is the first attempt to apply optimal transport theory to heterogeneous graph. By utilizing the optimal transport to calculate the transport plan between the meta-path view and aggregated view, it is no longer necessary to perform data augmentation and provide positive and negative samples for heterogeneous graph self-supervised learning.
- We propose a novel self-supervised Heterogeneous Graph neural network with Optimal Transport (HGOT) method, which utilize the optimal transport plan between the local semantics (branch view) and the global semantics (central view) of heterogeneous graphs to align the matching relationship between the graph space and the representation space, thereby obtaining higher-quality node representations.
- We conduct extensive experiments on four public benchmarks, demonstrating that the proposed HGOT outperforms the state-of-the-art methods. In particular, in the node classification task, HGOT achieves an average of more than 6% improvement in accuracy compared with state-of-the-art methods.

2. Related Work

Heterogeneous graph self-supervised learning. Heterogeneous graph self-supervised learning has been a promising paradigm, and it uses the information of the graph itself to learn node representations. For example, HeMuc (Zhang et al. 2023) proposes a contrastive learning method by designing a new sampling strategy that combines structure and characteristic information. HeCo (Wang et al. 2021a) applies two network structures to align both local schema and global metapath information. HGCML (Wang et al. 2023) constructs a contrastive learning method under multiple views by utilizing meta-paths and proposes a new positive sample sampling strategy to reduce sampling bias. In addition to contrastive learning, generative learning is also a good self-supervised learning method (Kipf and Welling 2016, Park et al. 2019). HGMAE (Tian et al. 2023) applies

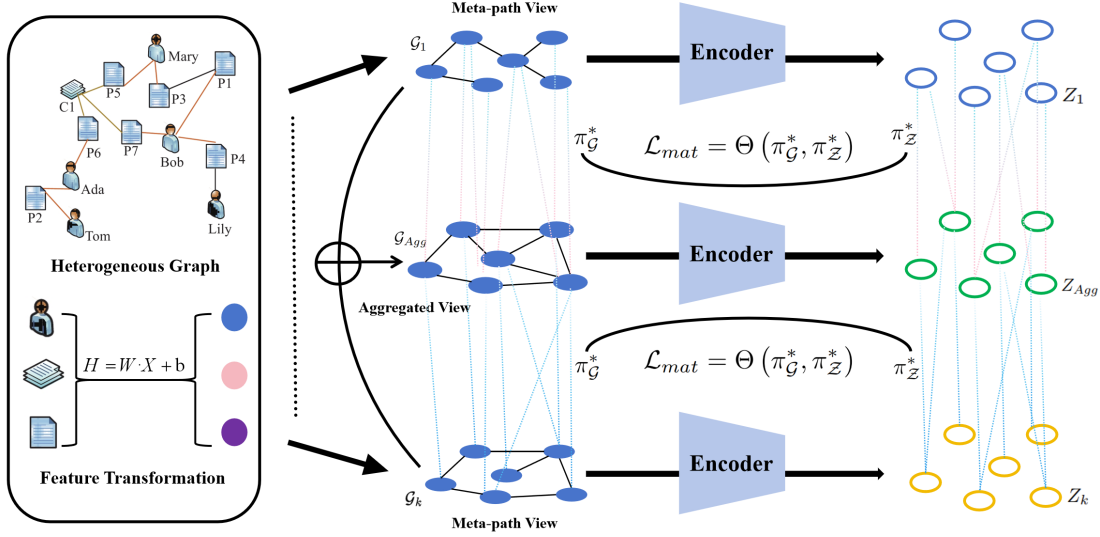


Figure 1. **The overall framework of HGOT:** We first project all nodes’ features of the original heterogeneous graph into the same dimensional space. We then design the heterogeneous semantic information aggregation to obtain the meta-path view and aggregated view. Finally, we introduce the optimal transport theory to discover matching relationships between each meta-path view and the aggregated view and align them to obtain higher quality node representations.

masked autoencoders to heterogeneous graphs and obtains better representations through the mask mechanism.

Optimal Transport. Optimal transport (OT) (Cédric Villani et al. 2009) is a mathematical tool widely used in computer vision (Nicolas Bonneel et al. 2011), generative adversarial network (Cao et al. 2019) and domain adaptation (Gu et al. 2022) to calculate the distance between two probability distributions or to align two distributions in different spaces. OT aims to derive a transport plan between a source and a target distribution, such that the transport cost is minimized. (Facundo Mémoli et al. 2011) proposes the Gromov-Wasserstein (GW) distance between metrics defined within each space which has been used as a distance between graphs in several applications (Vayer Titouan et al. 2020, Cédric Vincent-Cuaz et al. 2022, Xu et al. 2019). In this work, we combine the optimal transport theory with heterogeneous graph neural networks to form a new self-supervised learning paradigm.

3. Preliminaries

In this section, we define two concepts related to heterogeneous graphs and two problems of optimal transport on graphs as follows.

Definition 3.1. Heterogeneous Graph. A heterogeneous graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, T_{\mathcal{V}}, T_{\mathcal{E}}, \varphi, \psi)$, which consists of a node set \mathcal{V} , a link set \mathcal{E} , and it is associated with a node type mapping function $\varphi : \mathcal{V} \rightarrow T_{\mathcal{V}}$ and an edge type mapping function $\psi : \mathcal{E} \rightarrow T_{\mathcal{E}}$, where $T_{\mathcal{V}}$ and $T_{\mathcal{E}}$

denote sets of object and link types, and $|T_{\mathcal{V}}| + |T_{\mathcal{E}}| > 2$. It is a homogeneous graph with the same node and edge types when $|T_{\mathcal{V}}| = 1$ and $|T_{\mathcal{E}}| = 1$. For example, in the academic network ACM, three node types of Paper, Author, and Subject and two types of link relationships constitute a heterogeneous graph.

Definition 3.2. Meta-paths. A meta-path defines a composite relation of several edge types, represented as $\mathcal{P} = T_{\mathcal{V}_1} \xrightarrow{T_{\mathcal{E}_1}} T_{\mathcal{V}_2} \xrightarrow{T_{\mathcal{E}_2}} \dots \xrightarrow{T_{\mathcal{E}_r}} T_{\mathcal{V}_{(r+1)}}$ (abbreviated as $T_{\mathcal{V}_1} T_{\mathcal{V}_2} \dots T_{\mathcal{V}_{(r+1)}}$), which describes a composite relation $T_{\mathcal{E}} = T_{\mathcal{E}_1} \circ T_{\mathcal{E}_2} \circ \dots \circ T_{\mathcal{E}_r}$ between node types $T_{\mathcal{V}_1}$ and $T_{\mathcal{V}_{(r+1)}}$, where \circ denotes the composition operator on relations. Because meta-path is the combination of multiple relations, it contains high-order structures.

Problem 3.1. Plan and Optimal Transport. The optimal transport problem was first proposed by French mathematician Monge to seek the most cost-effective transportation solution for transporting the shape of one pile of sand into the shape of another (Wang et al. 2024). In particular, given two sets of features $X_1 = \{X_1^i\}_{i=1}^n$ and $X_2 = \{X_2^j\}_{j=1}^m$, where n and m are the number of features. $\mu \in \mathbb{R}^n$ and $\nu \in \mathbb{R}^m$ are the probability distributions of the entities in the two sets, respectively. It studies how to transform distribution μ into distribution ν with the minimum total transportation cost (i.e., the optimal transportation distance). This transport plan is called the optimal transport plan π , where the element of π describes the probability of moving mass from one position to another. In summary, the goal of optimal transport is to find a transport plan π that transforms

the distribution of node set X_1 into the distribution of node set X_2 with the minimum transport distance. In this work, we focus on the case of discrete distributions on graphs.

Problem 3.2. Optimal Transport for Graphs. On different graphs, we can view node attribute information as different distributions. If the nodes in the graph have no attributes or label information, we can also use the edge structure of the graph to calculate the optimal transportation plan between two graphs. Given two graphs $\mathcal{G}_1 = (\mathbf{A}_1, \mu)$ and $\mathcal{G}_2 = (\mathbf{A}_2, \nu)$, where \mathbf{A}_1 and \mathbf{A}_2 are the adjacency matrix of two graphs. Similarly, we can compute the optimal transport plan π between the edge structures (i.e., adjacency matrix) of the graph. The specific relevant formula for optimal transmission on the graph will be given in detail in the next section.

4. Method

In this section, we formally propose HGOT, a novel heterogeneous graph neural network with optimal transport. In particular, HGOT is constructed by three major components: node feature transformation, aggregated view generation, and multi-view optimal transport alignment. Figure 1 illustrates the framework of HGOT.

4.1. Node Feature Transformation

In a heterogeneous graph, different node types have features of different dimensions. Even if their feature vectors have the same dimensions, they may not be in the same feature space. Feature vectors of different dimensions are troublesome when we process them in a unified framework. In addition, considering that nodes are required to have the same dimensions when using the optimal transport calculation later, we project different types of node features into the same latent vector space.

Therefore, we apply a type specific linear transformation for each type of nodes by projecting feature vectors into the same latent factor space. Specifically, for the i -th node with type φ_i , we design a type-specific mapping matrix \mathbf{W}_{φ_i} to transform its features into a common latent space as follows:

$$\mathbf{h}_i = \mathbf{W}_{\varphi_i} \cdot \mathbf{x}_i + \mathbf{b}_{\varphi_i}, \quad (1)$$

where \mathbf{x}_i is the original feature vector, $\mathbf{h}_i \in \mathbb{R}^{d \times 1}$ is the projected feature of i -th node, and \mathbf{b}_{φ_i} denotes as vector bias.

The node feature transformation addresses the heterogeneity of a graph that originates from node features that are not in the same feature space. All nodes' projected features share the same dimension and are in the same feature space, which facilitates the calculation process of the part of the optimal transport module.

4.2. Aggregated View Generation

Since heterogeneous graphs contain rich semantic information which are reflected by various meta-paths, we design the aggregated view generation mechanism to aggregate different meta-path views. In particular, we decompose the original heterogeneous graph into multiple homogeneous meta-specific graphs according to meta-paths. Then we consider aggregating these homogeneous meta-specific graphs from the perspective of nodes and edges.

For nodes, following most heterogeneous GNNs' approaches, we first generate multiple views and each view corresponding to one meta-path that encodes one aspect of information. Then we leverage an attentive network to compute specific meta-path embedding $\tilde{\mathbf{h}}_i^p$ for node v_i under the p -th meta-path as

$$\tilde{\mathbf{h}}_i^p = \text{CONCAT}_{k=1}^K \Psi \left(\sum_{v_j \in \mathcal{N}_p(v_i)} \alpha_{ij}^p \mathbf{W}^p \mathbf{h}_j \right), \quad (2)$$

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

$$\alpha_{ij}^p = \frac{\exp \left(\Psi \left(\mathbf{a}_p^\top \left[\tilde{\mathbf{h}}_i^p \parallel \tilde{\mathbf{h}}_j^p \right] \right) \right)}{\sum_{v_k \in \mathcal{N}_p(v_i)} \exp \left(\Psi \left(\mathbf{a}_p^\top \left[\tilde{\mathbf{h}}_i^p \parallel \tilde{\mathbf{h}}_k^p \right] \right) \right)}, \quad (3)$$

where $\mathbf{a}_p \in \mathbb{R}^{2d \times 1}$ is a trainable linear weight vector for the p -th meta-path.

Finally, we fuse the node representations in all views into an aggregated representation. We employ another attentive network to obtain the aggregated representation $\mathbf{h}_i^{\text{agg}}$ that combines information from every semantic space by

$$\mathbf{h}_i^{\text{agg}} = \sum_{p=1}^{|\mathcal{P}|} \beta^p \cdot \tilde{\mathbf{h}}_i^p. \quad (4)$$

The coefficients are given by

$$\beta^p = \frac{\exp(\omega^p)}{\sum_{p' \in \mathcal{P}} \exp(\omega^{p'})}, \quad (5)$$

$$\omega^p = \frac{1}{|\mathcal{V}|} \sum_{v_i \in \mathcal{V}} \mathbf{q}^\top \cdot \tanh \left(\mathbf{M} \tilde{\mathbf{h}}_i^p + \tilde{\mathbf{b}} \right), \quad (6)$$

where $\mathbf{q} \in \mathbb{R}^{d_m}$ denotes type-level attention vector. β^p can be interpreted as the contribution of the p -th meta-path. $d_m \in \mathbb{R}$ is a hyperparameter, $\mathbf{M} \in \mathbb{R}^{d_m \times d}$, $\tilde{\mathbf{b}} \in \mathbb{R}^{d_m}$ are

the weight matrix and the bias vector, respectively. Finally, we concatenate and arrange the aggregated representations of all nodes and use a matrix \mathbf{Z}_{agg} to represent the representations of all nodes in the aggregated view.

For edges, we create the meta-path-based adjacency matrix \mathbf{A}_p for each meta-path $p \in \mathcal{P}$ via meta-path sampling. In order to directly obtain the edge structure information in the fused view, and do not want to introduce repeated edges, we perform a logical OR operation on all adjacency matrices as follows:

$$\mathbf{A}_{\text{agg}} = \mathbf{A}_1 \vee \mathbf{A}_2 \vee \dots \vee \mathbf{A}_{|\mathcal{P}|}, \quad (7)$$

where \mathbf{A}_{agg} denotes the adjacency matrix of aggregated view. Note that the operation rule here is to perform an OR operation on each element in the adjacency matrix. This obviously means that if two nodes are connected in any meta-path view, then there should be an edge between them in the aggregated view.

In this step, we obtain the node representation information \mathbf{Z}_{agg} and the edge structure information \mathbf{A}_{agg} of the aggregated view \mathcal{G}_{agg} .

4.3. Multi-view Optimal Transport Alignment

In order to achieve information matching from subgraphs induced by meta-paths to aggregated graph, we use the theory of optimal transport to calculate the optimal transport plan between each meta-path view and the aggregated view. Then we align the optimal transport plan between graph and representation to achieve information matching.

Our goal is to find the optimal transport plan for the two views, minimizing the transport cost (i.e., optimal transport distance). Therefore, we choose to start with the transport distance to calculate the optimal transport plan. Specifically, we obtain the node feature matrix \mathbf{H}_p and \mathbf{H}_{agg} in the meta-path view and the aggregated view after projection in the ‘‘node feature transformation’’ step. The formulation of the OT distance is

$$\mathcal{D}_n(\mathbf{H}_p, \mathbf{H}_{\text{agg}}) = \min_{\pi \in \Pi(\mu, \nu)} \sum_{i \in [n]} \sum_{j \in [m]} \mathcal{C}_X(\mathbf{H}_p^i, \mathbf{H}_{\text{agg}}^j) \cdot \pi_{ij}, \quad (8)$$

abbreviated as

$$\mathcal{D}_n(\mathbf{H}_p, \mathbf{H}_{\text{agg}}) = \min_{\pi \in \Pi(\mu, \nu)} \langle \mathcal{F}(\mathbf{H}_p, \mathbf{H}_{\text{agg}}), \pi \rangle, \quad (9)$$

where

$$\Pi(\mu, \nu) = \{\pi \in \mathbb{R}^{n \times m} | \pi \cdot \mathbf{1}_m = \mu, \mathbf{1}_n \cdot \pi = \nu\} \quad (10)$$

denotes all the joint distributions π with the marginal distribution μ and ν . $\mathcal{F}(\mathbf{H}_p, \mathbf{H}_{\text{agg}})_{ij} = \mathcal{C}_X(\mathbf{H}_p^i, \mathbf{H}_{\text{agg}}^j)$ is the

cost of moving \mathbf{H}_p^i to $\mathbf{H}_{\text{agg}}^j$, we can choose the cosine distance between them to calculate. The $\mathbf{1}$ denotes a vector that elements are all 1, $[n] = \{1, 2, \dots, n\}$, and $\langle \cdot, \cdot \rangle$ is the inner product operator. The $\pi \in \mathbb{R}^{n \times m}$ is called as transport plan. This distance $\mathcal{D}_n(\cdot, \cdot)$ is also known as the Wasserstein distance, used to calculate the transport distance between node distribution.

For graph, it is difficult to measure the cost between two nodes on different graphs without node label (attribute). Even if there is some way to get the cost between nodes, there is no edge information in the above Wasserstein distance formula. Therefore, we need to take the edge information into account as follows:

$$\mathcal{D}_e(\mathbf{A}_p, \mathbf{A}_{\text{agg}}) = \min_{\pi \in \Pi(\mu, \nu)} \sum_{i, j \in [n]^2} \sum_{k, l \in [m]^2} \mathcal{C}_A(\mathbf{A}_p^{ik}, \mathbf{A}_{\text{agg}}^{jl}) \cdot \pi_{ij} \pi_{kl}, \quad (11)$$

abbreviated as

$$\mathcal{D}_e(\mathbf{A}_p, \mathbf{A}_{\text{agg}}) = \min_{\pi \in \Pi(\mu, \nu)} \langle \mathcal{E}(\mathbf{A}_p, \mathbf{A}_{\text{agg}}) \otimes \pi, \pi \rangle, \quad (12)$$

where $\mathcal{E}(\mathbf{A}_p, \mathbf{A}_{\text{agg}})_{ijkl} = \mathcal{C}_A(\mathbf{A}_p^{ik}, \mathbf{A}_{\text{agg}}^{jl})$ and it is a 4-D tensor. $[n]^2 = [n] \times [n]$. Since the cost function here involves the adjacency matrix, we directly subtract it and then take the absolute value to calculate, which is $\mathcal{C}_A(\mathbf{A}_p^{ik}, \mathbf{A}_{\text{agg}}^{jl}) = |\mathbf{A}_p^{ik} - \mathbf{A}_{\text{agg}}^{jl}|$. This distance $\mathcal{D}_e(\cdot, \cdot)$ is also known as the Gromov-Wasserstein distance, used to calculate the transport distance between edge distribution. Then we consider the complete graph including node features and edge structure $\mathcal{G}_p = (\mathbf{H}_p, \mathbf{A}_p, \mu)$ and $\mathcal{G}_{\text{agg}} = (\mathbf{H}_{\text{agg}}, \mathbf{A}_{\text{agg}}, \nu)$. The fused Gromov-Wasserstein distance between two graphs can be defined as

$$\mathcal{D}_g(\mathcal{G}_p, \mathcal{G}_{\text{agg}}) = \min_{\pi \in \Pi(\mu, \nu)} \sigma \sum_{ij} \mathcal{C}_X(\mathbf{H}_p^i, \mathbf{H}_{\text{agg}}^j) \cdot \pi_{ij}^{\mathcal{G}} + (1 - \sigma) \sum_{ijkl} \mathcal{C}_A(\mathbf{A}_p^{ik}, \mathbf{A}_{\text{agg}}^{jl}) \cdot \pi_{ij}^{\mathcal{G}} \pi_{kl}^{\mathcal{G}}, \quad (13)$$

which is equivalent to

$$\min_{\pi \in \Pi(\mu, \nu)} \langle \sigma \mathcal{F}(\mathbf{H}_p, \mathbf{H}_{\text{agg}}) + (1 - \sigma) \mathcal{E}(\mathbf{A}_p, \mathbf{A}_{\text{agg}}) \otimes \pi_{\mathcal{G}}, \pi_{\mathcal{G}} \rangle, \quad (14)$$

where $\sigma \in [0; 1]$ represents the trade-off parameter for adjusting nodes and edges. The formula is a fusion of Equations (8) and (11).

Optimal Transport Plan. For graphs and representation vectors, they are in different spaces and are essentially two different objects. Although it is challenging to compare two different objects in different spaces, the optimal transport

plan provides a solution. In this work, we integrate the node features and topological structure (i.e., edges) on the graph and design a new objective function to align the optimal transport plan between the graph space and the representation space. Specifically, the fused optimal transport plan $\pi_G^* \in \mathbb{R}^{n \times m}$ between the two graphs \mathcal{G}_p and \mathcal{G}_{agg} above can be defined as

$$\pi_G^* = \arg \min_{\pi_G \in \Pi(\mu, \nu)} \langle \sigma \mathcal{F}(\mathbf{H}_p, \mathbf{H}_{agg}) + (1 - \sigma) \mathcal{E}(\mathbf{A}_p, \mathbf{A}_{agg}) \otimes \pi_G, \pi_G \rangle, \quad (15)$$

with the fused Gromov-Wasserstein distance. By tuning the parameter σ , we can control the deviation of the learned optimal transport plan between nodes and edges.

Then we use the backbone model (e.g., HGNNs) to obtain the node representation \mathbf{Z}_p under each meta-path view. The node representation here contains the semantic information of the p -th meta-path, and the node representation \mathbf{Z}_{agg} in the aggregated view calculated in Section 4.2. We can use Equation (8) directly to calculate the optimal plan $\pi_Z^* \in \mathbb{R}^{n \times m}$ as

$$\pi_Z^* = \arg \min_{\pi_Z \in \Pi(\mu, \nu)} \langle \mathcal{R}(\mathbf{Z}_p, \mathbf{Z}_{agg}), \pi_Z \rangle, \quad (16)$$

where $\mathcal{R}(\mathbf{Z}_p, \mathbf{Z}_{agg})_{ij} = \mathcal{C}_Z(\mathbf{Z}_p^i, \mathbf{Z}_{agg}^j)$, \mathbf{Z}_p and \mathbf{Z}_{agg} denote the node representations corresponding to \mathcal{G}_p and \mathcal{G}_{agg} , respectively.

Optimal Transport Alignment. Obviously, if the node embedding learned by the encoder can capture the matching information with other graphs while preserving its own information, it may obtain better node representations. Therefore, we force the encoder to preserve the matching relationship in graph space by aligning the optimal transfer plan between two graphs with the optimal transport plan between their corresponding node embeddings. We minimize the difference between the optimal transport plans in the two spaces as the alignment loss as follows

$$\mathcal{L}_{mat} = \Theta(\pi_G^*, \pi_Z^*), \quad (17)$$

where $\Theta(\cdot, \cdot)$ denotes the discrepancy function, it can be any commonly used metric. For example, we can choose the Frobenius-norm as $\Theta(\cdot, \cdot) = \|\cdot - \cdot\|_F$

In addition, to guide the encoder to retain the structural information of the original image while learning node embedding, we correct the cost matrix $\mathcal{R}(\mathbf{Z}_p, \mathbf{Z}_{agg})$ between representations, which implies the implicit structure relationships between nodes, in the representation space as follow

$$\mathcal{L}_{str} = \Theta(\sigma \mathcal{F}(\mathbf{H}_p, \mathbf{H}_{agg}) + (1 - \sigma) \mathcal{E}(\mathbf{A}_p, \mathbf{A}_{agg}) \otimes \pi_G^*, \mathcal{R}(\mathbf{Z}_p, \mathbf{Z}_{agg})). \quad (18)$$

To this end, we define the overall heterogeneous graph transport alignment loss as

$$\mathcal{L} = \mathcal{L}_{mat} + \rho \mathcal{L}_{str}, \quad (19)$$

where ρ is the trade-off parameter.

4.4. Comparison with Graph Contrastive Learning

A common way of traditional contrastive objective is contrasting views at the node level. For the representations (positive views) h_i^1 and h_i^2 of same node i in two augmented graphs \mathcal{G}_1 and \mathcal{G}_2 , the pairwise contrastive loss can be defined as

$$\mathcal{L}_{com} = \sum_i \log \left(\frac{e^{\mathcal{I}(h_i^1, h_i^2)/\tau}}{e^{\mathcal{I}(h_i^1, h_i^2)/\tau} + \sum_{k \neq i} e^{\mathcal{I}(h_i^1, h_k^2)/\tau}} \right) \quad (20)$$

where \mathcal{I} is a discriminator that maps two views to an agreement (similarity) score, such as the inner product. τ denotes the temperature.

Although our objective function deviates from the contrastive loss, as outlined in Equation (20), we identify a striking resemblance in the foundational algorithmic principles governing both methodologies. In this examination, we concentrate on the node level. When the attributes and contextual information of a given node v_i , demonstrate similarity across diverse semantic perspectives, the associated cost for these views diminishes, nearly reaching zero. This reduction in cost leads to an optimal transport strategy that favors a high probability of aligning the representations of these nodes. Leveraging Equations (17) and (18), the HGOT model adeptly fine-tunes both the matching probabilities and costs linked to the corresponding node representations within the output domain. Consequently, this ensures that the representations of two analogous nodes are sufficiently congruent. In contrast, for two distinct nodes v_i and v_k , the costs linked to their respective node views across varying semantic viewpoints are relatively higher. This discrepancy necessitates an adjustment in the opposite direction, ensuring that the representations of two dissimilar nodes are sufficiently differentiated.

From the perspective of contrastive learning, the process described above is in harmony with its core objectives. Nevertheless, there is a notable disparity between graph transport alignment and graph contrastive learning. HGOT capitalizes on calibration signals sourced directly from the graph space. This unique calibration signal empowers us to circumvent the need to distinguish between positive and negative samples.

5. Experiments

In this section, we conduct extensive experiments to verify the effectiveness of the proposed HGOT. Specifically, we first demonstrate the advantages of our proposed model over other methods in different tasks, including node classification, node clustering, and visualization of node representations. Then, we conduct other experiments to verify the effectiveness of the modules in our method, including the

Table 1. Experimental results on node classification.

Metric	DBLP		ACM		IMDB		Yelp	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
GCN	91.71 \pm 0.6	90.84 \pm 0.7	88.39 \pm 0.6	88.21 \pm 0.6	54.64 \pm 0.7	51.81 \pm 0.7	72.16 \pm 0.6	52.55 \pm 0.4
GAT	91.93 \pm 0.7	91.16 \pm 0.7	87.16 \pm 0.3	87.37 \pm 0.3	56.97 \pm 0.4	52.93 \pm 0.5	72.74 \pm 0.6	53.02 \pm 0.7
HAN	92.02 \pm 0.8	91.71 \pm 0.8	90.55 \pm 1.1	90.62 \pm 1.0	58.55 \pm 1.4	54.37 \pm 1.1	72.30 \pm 0.9	53.56 \pm 0.8
GraphSAGE	74.05 \pm 8.6	73.86 \pm 8.4	60.72 \pm 3.5	56.59 \pm 6.8	41.33 \pm 7.1	40.71 \pm 6.9	63.19 \pm 4.7	51.97 \pm 3.2
GAE	90.95 \pm 0.3	90.08 \pm 0.3	65.71 \pm 2.2	61.67 \pm 2.9	47.62 \pm 3.1	46.91 \pm 2.9	71.33 \pm 1.6	52.34 \pm 1.7
DGI	90.35 \pm 0.8	89.19 \pm 0.9	80.15 \pm 3.2	80.03 \pm 3.3	50.56 \pm 4.0	49.77 \pm 3.8	72.04 \pm 2.2	52.11 \pm 2.2
DeepWalk	86.23 \pm 0.4	84.85 \pm 0.5	73.87 \pm 0.6	74.13 \pm 0.6	48.35 \pm 0.8	45.39 \pm 0.7	69.44 \pm 1.0	49.81 \pm 0.8
Metapath2vec	90.61 \pm 0.1	90.81 \pm 0.1	73.61 \pm 0.3	73.80 \pm 0.4	48.82 \pm 1.2	45.18 \pm 0.8	71.88 \pm 0.2	50.89 \pm 0.6
HERec	91.17 \pm 0.3	90.74 \pm 0.3	73.85 \pm 0.9	73.92 \pm 0.8	50.91 \pm 0.8	47.62 \pm 0.8	72.03 \pm 0.5	51.35 \pm 0.3
FeaCon-V	91.04 \pm 0.5	91.50 \pm 0.4	87.76 \pm 0.3	88.14 \pm 0.4	53.86 \pm 0.8	50.65 \pm 1.0	72.27 \pm 0.7	53.72 \pm 0.7
FeaCon-D	91.13 \pm 0.3	88.82 \pm 0.3	87.77 \pm 0.8	87.27 \pm 0.5	53.37 \pm 0.6	42.32 \pm 0.8	72.74 \pm 1.1	52.90 \pm 0.5
HDGI	91.24 \pm 0.4	90.45 \pm 0.4	88.46 \pm 1.1	88.44 \pm 0.9	59.83 \pm 0.8	50.12 \pm 0.8	72.18 \pm 0.4	52.79 \pm 0.3
HEAD	91.30 \pm 0.7	91.08 \pm 0.8	87.05 \pm 1.5	87.28 \pm 1.5	57.39 \pm 0.9	49.01 \pm 1.2	74.61 \pm 2.0	55.97 \pm 2.0
HeCo	92.03 \pm 0.2	91.37 \pm 0.3	88.79 \pm 0.5	89.04 \pm 0.3	57.35 \pm 0.8	52.51 \pm 1.0	72.48 \pm 0.5	53.18 \pm 0.5
HGBER-V	92.77 \pm 0.4	91.80 \pm 0.4	92.34 \pm 0.7	91.55 \pm 0.4	59.97 \pm 0.6	55.16 \pm 1.2	73.14 \pm 0.6	53.48 \pm 0.6
HGBER-D	92.93 \pm 0.2	91.38 \pm 0.3	88.79 \pm 0.5	89.03 \pm 0.5	57.33 \pm 0.8	52.56 \pm 1.1	72.41 \pm 0.5	53.16 \pm 0.4
HGCML	91.11 \pm 0.5	90.47 \pm 0.3	87.90 \pm 0.3	88.46 \pm 0.4	57.19 \pm 0.7	51.03 \pm 0.4	73.35 \pm 0.8	52.90 \pm 0.6
HGMAE	91.89 \pm 0.3	91.60 \pm 0.5	89.15 \pm 0.4	89.29 \pm 0.4	60.13 \pm 0.4	60.09 \pm 0.6	72.57 \pm 0.6	56.33 \pm 0.3
HGOT	95.66 \pm 0.4	95.14 \pm 0.3	94.49 \pm 1.0	94.60 \pm 0.8	62.71 \pm 1.3	62.34 \pm 0.6	77.58 \pm 0.8	59.12 \pm 0.4

ablation study and parameter sensitivity analysis.

5.1. Experimental Setup

Datasets. Four public datasets (DBLP, ACM, IMDB and Yelp) are used to demonstrate the effectiveness of the proposed method. The detailed information on datasets is provided in attached page.

Baselines. We compare the proposed HGOT with three categories of baselines: **supervised methods:** GCN (T. N. Kipf et al. 2016), GAT (P. Velićković et al. 2017), a **semi-supervised heterogeneous method:** HAN (Wang et al. 2019), and **unsupervised methods:** GraphSAGE (William L. et al. 2017), GAE (T. N. Kipf et al. 2016), DGI (Petar Velickovic et al. 2019), DeepWalk (B. Perozzi et al. 2014), Metapath2vec (Dong et al. 2017), HERec (Shi et al. 2019), FeaCon-V, FeaCon-D, HDGI (Ren et al. 2019), HEAD (Wang et al. 2021), HeCo (Wang et al. 2021a), HGBER (Liu et al. 2023), HGCML (Wang et al. 2023) and HGMAE (Tian et al. 2023).

Implementation Details. For baselines, we adhere to the settings described in their original papers and follow the setup in HeCo. For the proposed HGOT, we use HAN as the backbone model to obtain the node representation Z_p under each meta-path view. We search the learning rate from $1e-4$ to $5e-3$, tune the patience for early stopping from 5 to 20. The Adam optimizer (Diederik P. Kingma et al. 2015) is adopted for gradient descent. For all methods, we set the embedding dimension as 64 and report the mean and standard deviation of 10 runs with different random seeds. Among all the methods, the HGBER model has two variants (HGBER-V and HGBER-D) and we compare them both.

5.2. Node Classification

We evaluate different models for the node classification task and report their performances in Table 1. We use Micro-F1 and Macro-F1 as evaluation metrics. Results demonstrate the effectiveness of HGOT as it achieves the best performance over all these baselines. Compared with the homogeneous graph model, HGOT can capture the high-order semantic relationship between nodes to obtain a better node representation. Therefore, our method has a significant improvement in the accuracy of node classification tasks compared to homogeneous graph methods, and even improves by about 34% compared to GraphSAGE on the ACM dataset. Although methods such as HAN and Metapath2vec have utilized meta-paths, and HeCo has attempted a contrastive learning scheme between meta-paths and network modes, these methods do not study the matching information between meta-path view and aggregated view. Because our model HGOT attempts to learn the matching information from the graph space and representation space, the quality of node representation is improved. Thus our HGOT achieves the best classification performance.

5.3. Node Clustering

In this task, we utilize hierarchical clustering algorithm to the learned embeddings of all nodes and the results of the node classification are shown in Table 2. Hierarchical clustering can generate hierarchical clustering structures. Compared with the K-means clustering algorithm, it does not require the number of clusters to be specified in advance. The appropriate number of clusters can be selected as needed, and it can process various types of data. Indicators such as clustering accuracy (ACC), normalized mutual information (NMI), and adjusted rand index (ARI) (Xia et al. 2014) are used as metrics to assess the quality of the

Table 2. Experimental results on node clustering.

Dataset	DBLP			ACM			IMDB			Yelp		
Metric	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
GraphSAGE	69.59	51.50	36.40	56.71	29.20	27.72	29.10	0.81	0.53	58.04	29.83	31.36
GAE	83.09	72.59	77.31	58.93	27.42	24.49	36.64	0.97	0.62	60.90	32.07	36.41
DeepWalk	82.33	70.97	75.61	58.82	29.50	28.86	38.29	0.93	0.66	61.83	35.88	38.67
Metapath2vec	75.62	71.33	78.58	58.91	31.17	31.08	38.74	1.27	1.78	62.96	35.53	40.11
HERec	76.54	71.39	79.47	63.33	42.18	36.70	38.92	1.29	1.58	63.84	35.09	40.18
FeaCon-V	68.21	45.26	40.49	68.22	45.18	40.44	42.88	4.47	3.92	63.29	36.61	40.27
FeaCon-D	80.47	56.89	56.50	78.62	53.66	52.09	43.67	3.55	3.90	63.37	37.02	39.94
HDGI	88.12	71.58	76.64	71.80	54.43	49.61	44.34	3.72	4.19	63.89	37.52	40.16
HEAD	88.93	71.60	73.48	80.92	56.71	43.55	44.19	3.44	4.20	63.89	38.04	42.25
HeCo	90.33	74.51	80.17	88.72	56.87	56.94	44.73	4.58	4.36	65.60	37.61	42.37
HGBER-V	89.97	72.14	79.60	89.10	65.88	70.52	44.77	4.72	4.63	65.37	38.75	42.56
HGBER-D	88.45	71.82	78.59	89.30	65.99	70.71	45.12	4.88	4.69	65.82	37.50	42.19
HGCML	90.36	73.28	79.61	89.21	65.13	71.05	50.66	9.34	9.15	65.72	37.71	42.49
HGMAE	91.47	76.92	82.34	89.83	66.68	71.51	53.90	11.36	12.03	62.10	39.04	42.88
HGOT	93.41	78.05	84.00	89.77	66.12	71.94	55.80	14.28	13.04	66.23	39.15	43.07

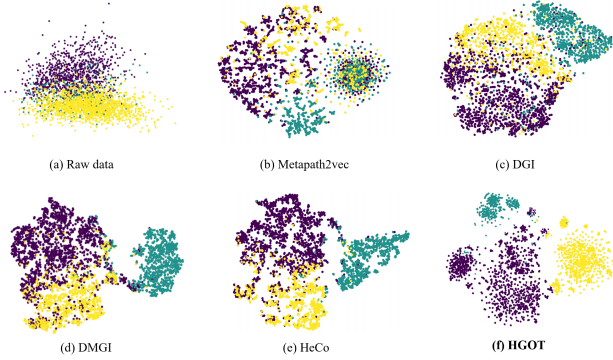


Figure 2. Visualization results of different methods on ACM.

clustering results. It can be seen that HGOT achieves the best performances on most datasets in node clustering task. Notably, HGOT demonstrates an average performance improvement of approximately 2.5% on the DBLP, IMDB, and Yelp datasets.

5.4. Visualization

In order to obtain more intuitive evaluation results, we conduct embedding visualization on ACM dataset. We use the t-SNE (Van Der Maaten L. 2014) dimensionality reduction method to experiment on different models and the results are shown in Figure 2, in which different colors mean different labels. As we can see, the Metapath2vec and DGI method does not classify nodes well, the visualization results show that they cannot make the boundaries between different types of nodes clearer because they cannot fuse all kinds of semantics. Although DMGI and HeCo have achieved the ability to clearly distinguish between various types of nodes, they are still mixed to some degree. The proposed HGOT correctly separates different nodes with relatively clear boundaries, demonstrating the effectiveness of HGOT.

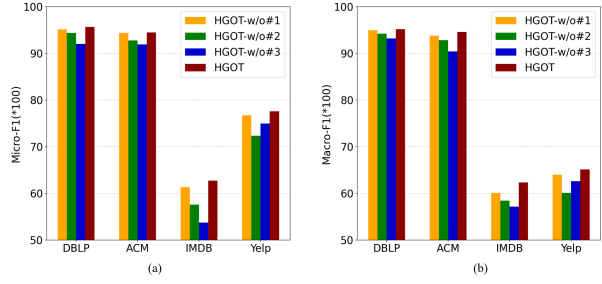


Figure 3. Ablation experimental results on four datasets.

5.5. Ablation Study

In this section, we compare HGOT with its variants on four datasets to verify the effectiveness of three components. HGOT-w/o#1 does not have the ‘‘aggregated graph generation’’ module. We directly match the two different meta-path views without calculating any information in the aggregated view. HGOT-w/o#2 represents HGOT without implicit structure loss \mathcal{L}_{str} . HGOT-w/o#3 drops the alignment of the optimal transport plan, but instead optimize the transport distance directly. In order to make the optimal transport distance between the graph space and the representation space consistent, we replace the matching loss with the following:

$$\mathcal{L}'_{mat} = |\mathcal{D}_g(\mathcal{G}_p, \mathcal{G}_{agg}) - \mathcal{D}_n(\mathbf{Z}_p, \mathbf{Z}_{agg})|, \quad (21)$$

where $|\cdot|$ denotes absolute value. The results are shown in Figure 3.

From Figure 3, conclusions can be drawn as follows: (1) The results of HGOT are consistently better than three variants, indicating the effectiveness of the heterogeneous semantic information aggregation, the implicit structure loss and the alignment of the optimal transport plan strategy. (2) HGOT-w/o#3 is the worst one, which makes us realize the necessity

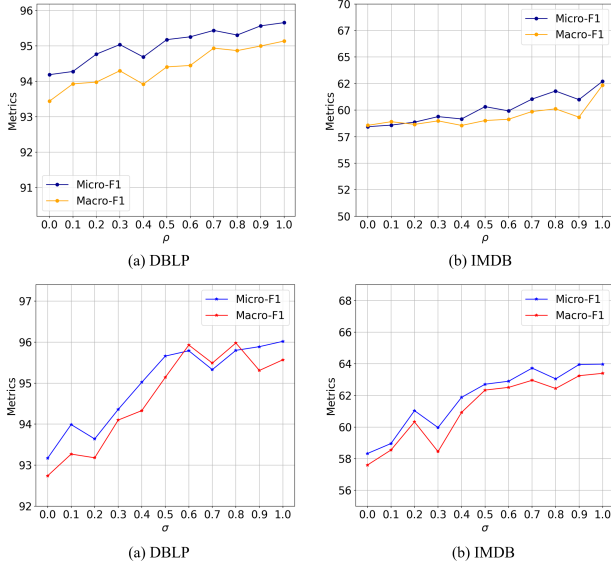


Figure 4. Analysis of hyperparameters ρ and σ on DBLP and IMDB datasets.

of the optimal transport plan alignment strategy. (3) The performance of HGOT-w/o#1 is also very competitive, which demonstrates that the strategy of aggregating all semantic information makes relatively little contribution to the overall model.

5.6. Parameter Sensitivity

In this section, we investigate the sensitivity of hyperparameters ρ and σ on two datasets. The experimental results are shown in Figure 4. We firstly fix the value of parameter σ to 0.5 and analyze the effect of parameter ρ on the prediction of performance. We can see that as the parameter ρ increases, HGOT shows an overall growth trend on both datasets. HGOT achieves optimal performance when $\rho = 1$. However, without the implicit structure loss, HGOT performs not well, which is also shown in the ablation study section. In addition, we fix the value of parameter ρ to 1, and analyze the effect of parameter σ on the prediction of performance. When $\sigma = 0$, HGOT only considers the matching information between edge structures, while when $\sigma = 1$, it only considers the matching information between nodes. Figure 4 reports the surprising outcome: With only node attributes for the calculation of the plan, the model achieves outstanding performance on both datasets. However, with both nodes and edges, the result is not optimal. At the same time, our model HGOT performs not well when only edges are considered. These experimental results give us an inspiration: In heterogeneous information networks, the number of edges may far exceed the number of nodes, so we can appropriately abandon some edge information. In this way, performance can be optimized while reducing the

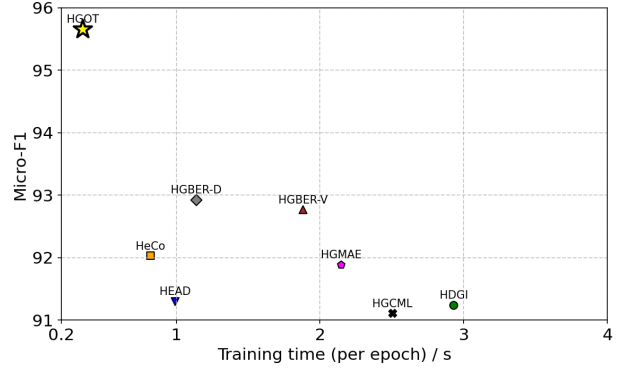


Figure 5. Micro-F1 scores and time consumption on DBLP dataset.

time complexity of the model.

5.7. Complexity Analysis

We select seven heterogeneous graph self-supervised learning methods and evaluate running time per training epoch on the DBLP dataset. According to the experimental result in Figure 5, we can see that HGOT has the best Micro-F1 score and requires the least training time, because it abandons the complex graph enhancement methods and the positive and negative sample selection process. The time complexity of the HGOT model mainly hinges on the optimization methods used for Equations (15) and (16). For Equation (15), which includes the fused Gromov-Wasserstein term, we use a conditional gradient (CG) solver for optimization. At each step of this solver, we need to calculate a gradient, and the time it takes for this calculation grows roughly in a cubic way as the number of nodes in the graph increases. On the other hand, Equation (16), which contains the Wasserstein term, can be optimized using the Sinkhorn-Knopp algorithm. This algorithm is quite efficient because its time complexity grows almost quadratically with the graph size.

6. Conclusion

In this paper, we propose a novel self-supervised heterogeneous graph neural networks with optimal transport, named HGOT. The attention mechanism is employed to obtain an aggregated view which can integrate the semantic information from different meta-paths. Then, HGOT exploits the optimal transport theory to discover the optimal transport plan between the meta-path view and the aggregated view. By aligning the transport plans between graph space and representation space, HGOT enforces the backbone model to learn node representations that precisely preserve the matching relationships. Extensive experiments conducted on multiple datasets demonstrate the state-of-the-art performance of the proposed HGOT.

Acknowledgements

This work is supported in part by National Natural Science Foundation of China (No. 62371340, 62172052, 62322203), Tianjin Natural Science Foundation Project (No. 24JCZDJC00820, 23JCYBJC00520), Tianjin Measurement Science and Technology Project (NO.2024TJMT061).

Impact Statement

This work provides a novel self-supervised learning approach for heterogeneous graphs in real-world scenario, and it aims to promote the development of heterogeneous graph neural networks in machine learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

- Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019. Heterogeneous Graph Attention Network. In *WWW*.
- Bai, Y.; Ying, Z.; Ren, H.; and Leskovec, J. 2021. Modeling heterogeneous hierarchies with relation-specific hyperbolic cones. In *NeurIPS*.
- Cao, Y.; Peng, H.; Wu, J.; Dou, Y.; Li, J.; and Yu, P. S. 2021. Knowledge-preserving incremental social event detection via heterogeneous gnns. In *WWW*.
- Wang, H.; and Leskovec, J. 2020. Unifying graph convolutional neural networks and label propagation. *arXiv preprint arXiv:2002.06755*.
- Dong, Y.; Hu, Z.; Wang, K.; Sun, Y.; and Tang, J. 2020. Heterogeneous Network Representation Learning. In *IJCAI*.
- Tian, Y.; Zhang, C.; Guo, Z.; Huang, C.; Metoyer, R.; and Chawla, N. V. 2022a. RecipeRec: A Heterogeneous Graph Learning Model for Recipe Recommendation. In *IJCAI*.
- Bansal, T.; Juan, D.-C.; Ravi, S.; and McCallum, A. 2019. A2N: Attending to neighbors for knowledge graph inference. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, 4387–4392.
- Wang, S.; Wei, X.; Nogueira dos Santos, C. N.; Wang, Z.; Nallapati, R.; Arnold, A.; Xiang, B.; Yu, P. S.; and Cruz, I. F. 2021a. Mixed-curvature multi-relational graph neural network for knowledge graph completion. In *WWW*, 1761–1771.
- Sun Y, Zhu D, Wang Y, et al. GTC: GNN-transformer co-contrastive learning for self-supervised heterogeneous graph representation. *Neural Networks*, 2025, 181: 106–115.
- Ji Z, Kong D, Yang Y, et al. ASSL-HGAT: Active semi-supervised learning empowered heterogeneous graph attention network. *Knowledge-Based Systems*, 2024, 290: 156-167.
- Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. 2020. Strategies for Pre-training Graph Neural Networks. In *ICLR*.
- Wang, X.; Liu, N.; Han, H.; and Shi, C. 2021a. Self-supervised heterogeneous graph neural network with co-contrastive learning. In *KDD*.
- Zhao Z, Zhu Z, Liu Y, et al. Heterogeneous Graph Contrastive Learning with Augmentation Graph. *IEEE Transactions on Artificial Intelligence*, 2024.
- Zhao M, Jia A L. Dahgn: Degree-aware heterogeneous graph neural network. *Knowledge-Based Systems*, 2024, 285: 135-145.
- Xie Y, Yu C, Jin X, et al. Heterogeneous graph contrastive learning for cold start cross-domain recommendation. *Knowledge-Based Systems*, 2024: 2054-2065.
- Liu Y, Xia L, Huang C. 2024. Selfgnn: Self-supervised graph neural networks for sequential recommendation. In *SIGIR*.
- Chen L, Cao J, Liang W, et al. Geography-aware Heterogeneous Graph Contrastive Learning for Travel Recommendation. *ACM Transactions on Spatial Algorithms and Systems*, 2024.
- Yang, C.; Xiao, Y.; Zhang, Y.; Sun, Y.; and Han, J. Heterogeneous Network Representation Learning: A Unified Framework with Survey and Benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 2020, 34(10): 4854-4873.
- Wang, X.; Bo, D.; Shi, C.; Fan, S.; Ye, Y.; and Philip, S. Y. A survey on heterogeneous graph embedding: methods, techniques, applications and sources. *IEEE Transactions on Big Data*, 2022, 9(2): 415-436.
- Fan, Y.; Ju, M.; Zhang, C.; and Ye, Y. 2022. Heterogeneous temporal graph neural network. In *SDM*.
- Fu, X.; Zhang, J.; Meng, Z.; and King, I. 2020. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *WWW*.
- Zhang Q, Zhao Z, Zhou H, et al. Self-supervised contrastive learning on heterogeneous graphs with mutual constraints of structure and feature. *Information Sciences*, 2023, 640: 119026.
- Kipf, T. N.; and Welling, M. 2016. Variational graph auto-

- encoders. *arXiv preprint arXiv:1611.07308*.
- Park, J.; Lee, M.; Chang, H. J.; Lee, K.; and Choi, J. Y. 2019. Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In *ICCV*.
- Wang Z, Li Q, Yu D, et al. 2023. Heterogeneous graph contrastive multi-view learning. In *SDM*.
- Tian Y, Dong K, Zhang C, et al. 2023. Heterogeneous graph masked autoencoders. In *AAAI*.
- Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.
- Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–12, 2011.
- Jiezhong Cao, Langyuan Mo, Yifan Zhang, Kui Jia, Chunhua Shen.2019. and Minghui Tan. Multi-marginal wasserstein gan. In *NeurIPS*.
- Xiang Gu, Yucheng Yang, Wei Zeng, Jian Sun, and Zongben Xu. 2022. Keypoint-guided optimal transport with applications in heterogeneous domain adaptation. In *NeurIPS*.
- Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 2011, 11:417–487.
- Vayer Titouan, Nicolas Courty, Romain Tavenard, and Rémi. 2019. Flamary. Optimal transport for structured data with application on graphs. In *ICML*.
- Cédric Vincent-Cuaz, Rémi Flamary, Marco Corneli, Titouan. 2022. Vayer, and Nicolas Courty. Semi-relaxed gromov-wasserstein divergence and applications on graphs. In *ICLR*.
- Hongteng Xu, Dixin Luo, and Lawrence Carin. 2019. Scalable gromov-wasserstein learning for graph partitioning and matching. In *NeurIPS*.
- Wang Y, Zhao Y, Wang D Z, et al. 2024. GALOPA: graph transport learning with optimal plan alignment. In *NeurIPS*.
- T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks, 2016, *arXiv:1609.02907*.
- P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks, 2017, *arXiv:1710.10903*.
- William L. Hamilton, Zitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS*.
- Thomas N. Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).
- Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *ICLR*.
- B. Perozzi, R. Al-Rfou, and S. Skiena, 2014. DeepWalk: Online learning of social representations. In *KDD*.
- Y. Dong, N. V. Chawla, and A. Swami, 2017. Metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*.
- C. Shi, B. Hu, W. X. Zhao, and P. S. Yu. Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, 2019.
- Y. Ren, B. Liu, C. Huang, P. Dai, L. Bo, and J. Zhang, Heterogeneous deep graph infomax. 2019, *arXiv:1911.08538*.
- R. Wang, C. Shi, T. Zhao, X. Wang, and Y. F. Ye, Heterogeneous information network embedding with adversarial disentangler. *IEEE Trans. Knowl. Data Eng.*, Jul. 13, 2021.
- Liu Y, Fan L, Wang X, et al. HGBER: Heterogeneous graph neural network with bidirectional encoding representation. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.
- R. Xia, Y. Pan, L. Du, and J. Yin. 2014. Robust multi-view spectral clustering via low-rank and sparse decomposition, In *AAAI*.
- Van Der Maaten L. Accelerating t-SNE using tree-based algorithms. *The Journal of Machine Learning Research*, 2014, 15(1): 3221-3245.

A. Datasets Details

The following four public datasets are used to verify the effectiveness of the proposed method.

DBLP: DBLP is an integrated database of English-language literature in the field of computer with the results of research as the author as the core. In this article, a subset of DBLP which consists of 14 328 papers (P), 4057 authors(A), 20 conferences(C), and 8789 terms(T) is extracted. We divide the authors into four categories according to their research areas: *database, data mining, machine learning, information retrieval*. Depending on the conferences linked to the authors, we classify the authors into the appropriate research areas. The attribute information of authors is extracted from keywords. Three meta-paths, APA, APCPA, and APTPA are extracted for use with experiments.

ACM: ACM is a paper network contrived from the ACM dataset using papers published in ACM Knowledge Discovery and Data Mining (KDD), SIGMOD, SIGCOMM, Mobi-COMM, and VLDB. ACM dataset consists of 3025 papers (P), 5835 authors (A), and 56 subjects (S). The papers are divided into three categories: *database, wireless communication, data mining*. Depending on the conferences linked to the papers, we classify the papers into the appropriate categories. The attribute information of papers is extracted from keywords. Two meta-paths PAP, PSP are extracted for use with experiments.

IMDB: IMDB is a link dataset built with permission from the Internet Movie Data (IMDB). A subset of IMDB which contains 4780 movies (M), 5841 actors (A), and 2269 directors (D) is extracted. We divide the movies into three categories: *Action, Comedy, Drama* according to their genre. The attribute information of movies is extracted from plots. Two meta-paths MAM , MDM are extracted for use with experiments.

Yelp: The Yelp dataset contains 2614 businesses (B), 1286 users (U), two reviews (R), two services (S), and nine rating levels (L). The business nodes are labeled by their category. The node features are constructed by the bag-of-words representation of the related keywords. Four meta-paths, BUB, BRB, BSB, and BLB, are extracted for use with experiments.

B. Dimension Analysis

We further perform dimension sensitivity analysis to show the impact of hidden dimensions in Figure 5. In particular, we search the number of hidden dimensions from $\{64, 128, 256, 512, 1024\}$, as shown the inflection point of the discount in the figure. It can be seen from the figure, the optimal hidden dimension can be different across different datasets. As shown in the Figure 6, the best hidden dimension on the DBLP dataset is 256, while on the IMDB is 512. In addition, as the hidden dimension increases, the model effect may be better. For example, when the hidden dimension is only 64 dimensions, the model effect is not good, because using a small hidden dimension could prevent the model from fully capturing the knowledge. However, too wide a hidden dimension could decentralize the model’s focus on meaningful information. Therefore, the model under the standard latent dimension will capture more complex and useful information, and the effect can be optimal.

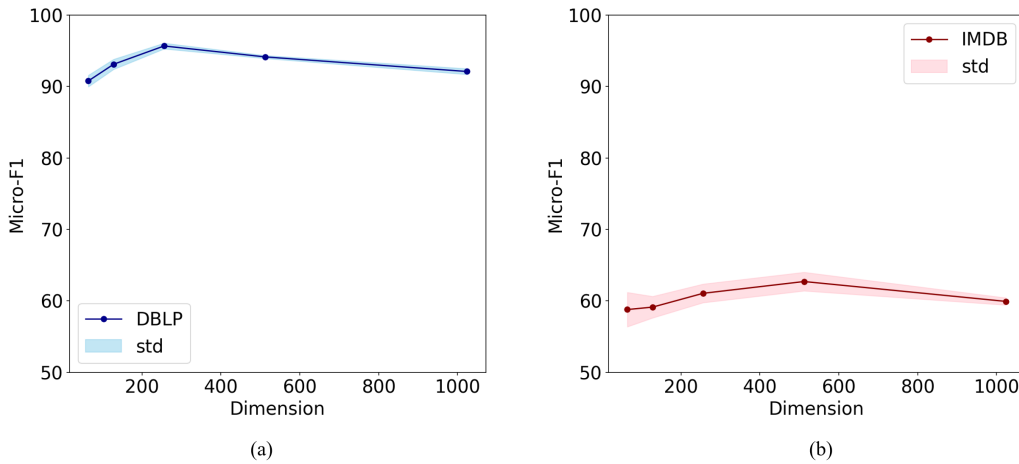


Figure 6. Performance of HGOT under different hidden dimensions.

C. General Settings

Environment. The environment in which we run experiments is:

- Operating system: Microsoft Windows 11 64-bit
- CPU information: 12th Gen Intel(R) Core(TM) i9-12900H (20 CPUs), 2.5GHz
- GPU information: NVIDIA GeForce RTX 3070 Ti Laptop GPU

Resources. The address and licenses of all datasets are as follows:

DBLP: <https://dblp.uni-trier.de>

ACM: <http://dl.acm.org/>

IMDB: <http://komarix.org/ac/ds/>

Yelp: <https://www.yelp.com/dataset>