WAGE: Weight-Sharing Attribute-Missing Graph Autoencoder

Wenxuan Tu, Sihang Zhou, *Member, IEEE*, Xinwang Liu, *Senior Member, IEEE*, Zhiping Cai, Yawei Zhao, Yue Liu, *Member, IEEE*, and Kunlun He

***Abstract*—Attribute-missing graph learning, a common yet challenging problem, has recently attracted considerable attention. Existing efforts have at least one of the following limitations: 1) lack a noise filtering and information enhancing scheme, resulting in less comprehensive data completion; 2) isolate the node attribute and graph structure encoding processes, introducing more parameters and failing to take full advantage of the two types of information; and 3) impose overly strict distribution assumptions on the latent variables, leading to biased or less discriminative node representations. To tackle the issues, based on the idea of introducing intimate information interaction between the two information sources, we propose Weight-sharing Attribute-missing Graph autoEncoder (WAGE) to boost the expressive capacity of node representations for high-quality missing attribute reconstruction. Specifically, three strategies have been conducted. Firstly, we entangle the attribute embedding and structure embedding by introducing a weight-sharing architecture to share the parameters learned by both processes, which allows the network training to benefit from more abundant and diverse information. Secondly, we introduce a K-nearest neighbor-based dual non-local learning mechanism to improve the quality of data imputation by revealing unobserved high-confidence connections while filtering unreliable ones. Thirdly, we manually mask the connections on multiple adjacency matrices and force the structure-oriented embedding subnetwork to recover the actual adjacency matrix, thus enforcing the resulting network to be able to selectively exploit more high-order discriminative features for data completion. Extensive experiments on six benchmark datasets demonstrate the effectiveness and superiority of WAGE against state-of-the-art competitors.**

***Index Terms*—Attribute-missing, graph autoencoder, multi-view learning, unsupervised graph learning.**

Received 7 June 2023; revised 27 December 2024; accepted 13 March 2025. Date of publication 24 March 2025; date of current version 6 June 2025. This work is supported by the National Natural Science Foundation of China under Grant 62441618, Grant 62325604, and Grant 62276271, in part by the Natural Science Foundation of Hainan University under Grant XJ2400009401, and in part by the funding Grant 22-TQ23-14-ZD-01-001. Recommended for acceptance by K. M. Lee. *(Wenxuan Tu and Sihang Zhou contributed equally to this work.) (Corresponding authors: Xinwang Liu; Kunlun He.)*

Wenxuan Tu is with the School of Computer Science and Technology, Hainan University, Haikou 570228, China (e-mail: wenxuantu@163.com).

Sihang Zhou is with the School of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: sihangjoe@gmail.com).

Xinwang Liu, Zhiping Cai, and Yue Liu are with the School of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: xinwangliu@nudt.edu.cn; zpcai@nudt.edu.cn; yueliu19990731@163.com).

Yawei Zhao and Kunlun He are with Medical Big Data Research Center, Chinese PLA General Hospital, Beijing 100853, China (e-mail: csyawei.zhao@gmail.com; kunlunhe@plagh.org).

Digital Object Identifier 10.1109/TPAMI.2025.3554053

I. Introduction

GRAPH representation learning (GRL), which aims to learn a graph neural network (GNN) that embeds graph data into a low-dimensional latent space by preserving node attributes and graph structures, has been intensively studied and widely applied to various applications [1], [2], [3], [4], [5], [6], [7], [8]. One underlying assumption commonly adopted by these methods is that all attributes of nodes are complete and trustworthy. However, in practice, this assumption may not hold due to: 1) the absence of a portion of attributes within all nodes; and 2) the absence of all attributes of particular nodes. These circumstances are called attribute-incomplete [9], [10] and attribute-missing [11], [12] problems, respectively. The existence of the above circumstances makes existing graph learning methods unable to handle corresponding learning problems effectively.

To solve the first type of problem, the early methods mainly concentrate on developing imputation techniques for data completion, such as matrix completion via matrix factorization [13], [14], [15] and generative adversarial network [9], [16]. These imputation-based methods generally disconnect the data imputation and network learning procedures, which decreases the diversity and discriminability of learned node representations. To solve the problem, the following methods, such as GRAPE [17] and GCNMF [10], integrate both data imputation and representation learning processes into a united GNN-based optimization framework. Specifically, in these methods, incomplete attributes are carefully recovered by employing a bipartite message-passing strategy and a Gaussian mixture model (GMM)-based learning scheme, respectively. Although these methods have proven effective in addressing the attribute-incomplete problem, the proposed techniques for recovering attribute-incomplete samples would fail in scenarios where node attributes are entirely missing. As a result, their abilities to achieve high-quality data completion for attribute-missing samples are often compromised, posing a significant challenge to their overall performance.

Compared to the first type of method, another series of research aims to tackle the attribute-missing problem. Unlike the situation where attributes are incomplete, in the attribute-missing case, a portion of samples within a graph only exhibit local topology without any node features. To address this issue, a commonly adopted solution is to recover the missing data from observed attributes with the aid of complete graph structures. For example, SAT [11] makes the first attempt to guide the generation of meaningful latent features by introducing a distribution assumption between attribute embedding and structure embedding sub-networks. Inspired by its success, recent advances have been devoted to handling attribute-missing graph learning tasks by inheriting the main idea of decoupled structure-attribute embedding [12], [18] and data distribution assumption [19], [20]. Despite demonstrating high-quality data completion across multiple tasks, current attribute-missing graph learning methods suffer from at least one of the following non-negligible limitations: 1) lack a noise filtering and information enhancing scheme for missing attribute estimation. Most studies complete missing attributes in the noisy latent space and do not verify the topology relationships of reconstructed attribute-missing samples, making the data completion procedure less comprehensive; 2) isolate the node attribute and graph structure encoding processes. Besides introducing more parameters, the decoupled encoding design usually struggles to fully merge and harness two-source heterogeneous information; and 3) impose overly strict distribution assumptions on the latent variables. This would make it difficult for the model to accurately represent the true distribution of the data, leading to biased or inaccurate inference for attribute-missing samples.

The above observations motivate us to propose a novel attribute-missing graph learning method termed **W**eight-sharing **A**ttribute-missing **G**raph auto**E**ncoder (WAGE). The core idea behind WAGE is to establish a structure-attribute mutually enhanced learning strategy to promote the accurate reconstruction of node attributes and local topology for attribute-missing samples. To this end, we tightly entangle the node attribute and graph structure encoding processes by designing a weight-sharing architecture that shares the learned patterns from different views. Then, we design a dual non-local aggregating (DNA) module that leverages a K-nearest neighbor-based dual non-local learning mechanism to identify high-confidence connections while filtering unreliable ones. By doing this, each attribute-missing node in the latent space could effectively collect and preserve the most informative non-local features, acquiring more reliable information for data imputation. Moreover, to model accurate topology relationships between reconstructed attribute-missing nodes, we introduce a hidden structure enhancing (HSE) module to assist the DNA module in regularizing the learned node representation with the enhanced topology information. This module involves masking connections on multiple adjacency matrices and compelling the embedding sub-network to recover the actual adjacency matrix. As an auxiliary task, the HSE module facilitates the network to narrow the distance between neighbor sample representations and enhance their correlations, especially for attribute-missing nodes. Consequently, the resulting network could selectively utilize more distinctive features from hidden high-order attributes for data imputation. Finally, through the iterative optimization of the two-source information encoding, DNA module, and HSE module in an end-to-end training process, the interdependence of these components can be leveraged to enhance the discriminative power of the graph embedding for high-quality data completion. The main contributions of this paper are listed as follows:

- We propose a novel unsupervised attribute-missing graph learning method termed WAGE, which follows the filtering-and-enhancing data completion principle. The advantages of WAGE over state-of-the-art methods can be attributed to its more comprehensive data completion, more compact data-completing architecture, and free of prior assumptions.
- We design a novel DNA module to estimate missing attributes in our data completion strategy. The module is able to reveal unobserved high-confidence connections while filtering unreliable ones, preserving more reliable clues during data imputation. From the aspect of sample correlation analysis, we develop a new HSE module to enhance the topology relationships of attribute-missing samples, which can help accurately estimate missing attributes.
- Extensive experiments on six benchmark datasets have demonstrated the superiority of WAGE and the effectiveness of each component. Our method can yield highly competitive or better performance compared to state-of-the-art competitors.

The remainder of this paper is organized below. Section II provides a comprehensive review of related works. Section III presents the notations and definitions, introduces the model, and outlines the training objectives. Section IV reports experimental results along with corresponding analyses. Section V highlights the distinctions between WAGE and several representative graph learning methods. Finally, Section VI concludes the paper and discusses future directions.

II. Related Work

A. Graph Representation Learning

After achieving remarkable performance in computer vision [21], [22] and natural language processing [23], [24] domains, representation learning on graphs has emerged as a competitive and increasingly popular approach in graph machine learning. The early graph representation learning (GRL) methods focus on learning node representations by utilizing probability models on random walk paths generated on graphs, such as DeepWalk [25] and node2vec [26]. However, these methods tend to overly emphasize the structural information while ignoring the equally important attribute information. Thanks to the advancement of graph neural networks (GNNs), GNN-based representation learning methods have emerged, which effectively exploit both graph structure information and node attribute information in either a spectral [27] or spatial [28] domain. These methods have garnered significant attention and have been extensively researched in recent years. As one of the most representative approaches, generative/predictive GRL methods explore rich information from data itself via various deep learning techniques, such as autoencoder learning and generative adversarial learning [29], [30], [31], [32], [33], [34], [35], [36]. Another research line is contrastive GRL, which focuses on maximizing the agreement of two jointly sampled positive pairs [37], [38], [39], [40], [41], [42], [43], [44], [45]. One common underlying assumption adopted by current graph learning methods is that all node attributes are complete and trustworthy. However, in practice, the absent data makes it challenging for existing graph learning methods to achieve satisfactory performance.

B. Deep Graph Learning With Absent Data

Based on the degree of data absence, absent graphs can be broadly classified into two categories: attribute-incomplete graphs and attribute-missing graphs. To process attribute-incomplete graphs, one popular way commonly adopted by existing graph learning methods is data imputation, which can be achieved by matrix completion [46], [47], generative adversarial network [9], Gaussian mixture model [10], feature propagation [48], and self-distillation [49]. For matrix completion, GC-MC [46], NMTR [15], and IGMC [47] first formulate the user-item rating matrix, users (or items), and the observed ratings as bipartite graph, nodes, and links, respectively. Then they apply a graph neural network (GNN) to predict the absent linkages between node pairs for data completion in a transductive or inductive manner. Moreover, GINN [9] first initializes the incomplete values by a binary mask matrix before network training and then introduces an adversarial loss to train a denoising autoencoder for data completion. To further enhance the quality of missing attribute estimation, recent efforts integrate data imputation and representation learning processing into a united GNN-based framework. For example, GRAPE [17] formulates the feature imputation as an edge-level prediction task on the graph and employs a GNN to solve it. GCNMF [10] utilizes a Gaussian mixture model (GMM) to estimate incomplete attributes by making them conform with a Gaussian mixture distribution. FP [48] diffuses the features from observed nodes to neighbors whose features are incomplete based on the heat diffusion equation. More recently, a general teacher-student graph learning framework termed T2-GNN [49] is designed to restore incomplete node features and graph structures through self-distillation.

Although the aforementioned methods are capable of effectively addressing the attribute-incomplete problem, they may encounter challenges in processing graphs where a certain portion of nodes have entirely missing attributes. This issue has attracted significant attention from researchers in the field of deep graph learning [12], [18], [19], [20], [48], [50], [51], [52]. For example, HGNN-AC [50] employs heterogeneous information networks (HINs) to extract node topological representations and takes the sample correlation as learning guidance to conduct an attention mechanism-based feature completion for attribute-missing samples. HGCA [51] unifies the processes of feature completion and representation learning, and leverages contrastive learning to conduct a fine-grained attribute completion by extracting the semantic relations among different types of samples. Besides heterogeneous attribute-missing graphs, an advanced method termed SAT [11] makes the first attempt to solve the attribute-missing problem in homogeneous graphs under the guidance of a shared-latent space assumption. Likewise, CSAT [18] further improves SAT by conducting a topology distribution contrast in the latent space. In other recent works, ITR [12] introduces an initializing-then-refining mechanism, which enables the network to fully use the trustworthy visible information to adaptively conduct the sample embedding for missing attribute estimation. PSGNN [52] studies the usage of artificial positional node features and structural node features to help GNNs learn useful information on non-attributed graphs. More recently, several auto-encoder-style frameworks such as SVGA [19] and Amer [20] are developed to complete missing node features via structured variational inference and generative adversarial learning, respectively.

As previously discussed, although these methods have demonstrated success, they suffer from at least one of the following major limitations: 1) less comprehensive data completion; 2) structure-attribute embedding isolation; and 3) overly strict distribution assumptions. In contrast, WAGE advances most attribute-missing graph learning methods by facilitating the mutual enhancement of both structure and attribute information, which enjoys the following merits: more comprehensive data completion, more compact data-completing architecture, and free of any prior assumptions. In the Discussion Section, we thoroughly analyze the differences between WAGE and some representative methods, including GraphMAE [53], T2-GNN [49], SAT [11], Amer [20], CSAT [18], ITR [12], FP [48], and PaGNN [54].

III. Method

A. Notations and Definitions

Given an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with C categories, where $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^{N}$, $\mathcal{E}$, and N are the node set, edge set, and the number of nodes, respectively. In classic graph learning, a graph is usually characterized by attribute matrix $\mathbf{X} \in \mathbb{R}^{N\times D}$ and normalized adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ [28], where D refers to node attribute dimension. Table I summarizes the frequently used notations.

Definition 1 (Attribute-missing graph): In attribute-missing graph learning, with the existence of missing attributes, we define $\mathcal{V}^o = \{\mathbf{v}_g^o\}_{g=1}^{N^o}$ and $\mathcal{V}^m = \{\mathbf{v}_k^m\}_{k=1}^{N^m}$ to form the set of attribute-observed nodes and the set of attribute-missing nodes, respectively. Accordingly, $\mathcal{V} = \mathcal{V}^o \cup \mathcal{V}^m$, $\mathcal{V}^o \cap \mathcal{V}^m = \varnothing$, and $N = N^o + N^m$. To ease the model training under a high missing ratio r, we fill the missing attributes with a set of initial values $\mathcal{T}^m = \{\mathbf{t}_k^m\}_{k=1}^{N^m}$, where $\mathbf{t}_k^m \in \mathbb{R}^D$ could represent zero values or observed neighbor values. Based on these notations, an attribute-missing graph could be formulated as $\widetilde{\mathcal{G}} = \{\mathcal{V}^o, \mathcal{T}^m, \mathcal{E}\}$.

Definition 2 (Edge-masked attribute-missing graph): Since the structure information of $\mathcal{G}$ is complete and trustworthy by default, to fully exploit diverse sample correlations, we construct a multi-order edge set $\mathcal{E}_M = \{\mathcal{E}^{h\text{-}th}\}_{h=1}^{H}$ using a random walk-like operation, where $H = |\mathcal{E}_M|$ is the number of neighbor orders. Specially, we formulate $\mathcal{E}^{h\text{-}th}$ as a $h\text{-}th$-order adjacency matrix $\mathbf{A}^{h\text{-}th} \in \mathbb{R}^{N\times N}$, which can be calculated by:

$$\mathbf{A}^{h\text{-}th} = \mathbf{A}^{1st}\mathbf{A}^{(h-1)\text{-}th}, \tag{1}$$

where $\mathbf{A}^{1st}$ denotes the raw adjacency matrix of a given graph and $\mathbf{A}^{0\text{-}th}$ here refers to an identity matrix $\mathbf{I} \in \mathbb{R}^{N\times N}$. $\mathbf{A}_{ij}^{h\text{-}th} =$ 1 if node $\mathbf{v}_i$ and node $\mathbf{v}_j$ are connected, $\mathbf{A}_{ij}^{h\text{-}th} = 0$ otherwise. It is worth noting that the random walk-based methods usually suffer from a common issue, *i.e.,* node revisiting. To eliminate

TABLE I
SUMMARY OF FREQUENTLY USED NOTATIONS

Notations	Meaning
$\widetilde{\mathcal{G}}$	Attribute-missing graph
$\widetilde{\mathcal{G}}^{1st}, \widetilde{\mathcal{G}}^{2nd}, \ldots, \widetilde{\mathcal{G}}^{H\text{-}th}$	Edge-masked attribute-missing graphs
$\mathcal{E}$	Edge set
$\mathcal{E}_M, \widetilde{\mathcal{E}}_M$	Multi-order edge/masked edge set
$\mathcal{V}$	All node set
$\mathcal{V}^o, \mathcal{V}^m$	Attribute-observed/attribute-missing node set
$\mathcal{T}^m$	Initial value set
N	The number of all nodes
N^o	The number of attribute-observed nodes
N^m	The number of attribute-missing nodes
C	The number of data categories
D, d	Node attribute/representation dimension
r	Attribute-missing ratio
$\mathbf{X} \in \mathbb{R}^{N \times D}$	Original attribute matrix
$\widetilde{\mathbf{X}} \in \mathbb{R}^{N \times D}$	Initially imputed attribute matrix
$\mathbf{A} \in \mathbb{R}^{N \times N}$	Normalized adjacency matrix
$\mathbf{A}^{h\text{-}th} \in \mathbb{R}^{N \times N}$	$h\text{-}th$-order adjacency matrix
$\widetilde{\mathbf{A}}^{h\text{-}th} \in \mathbb{R}^{N \times N}$	Edge-masked $h\text{-}th$-order adjacency matrix
$\mathbf{Z} \in \mathbb{R}^{N \times d}$	Node representation matrix of $\widetilde{\mathcal{G}}$
$\mathbf{S} \in \mathbb{R}^{N \times N}$	Affinity matrix
$\mathbf{S}_G \in \mathbb{R}^{N \times N}$	Global-scope indicator matrix
$\mathbf{S}_L \in \mathbb{R}^{N \times N}$	Multi-order local-scope indicator matrix
$\mathbf{Z}_A \in \mathbb{R}^{N \times d}$	Attribute-enhanced representation matrix
$\mathbf{Z}^{h\text{-}th} \in \mathbb{R}^{N \times d}$	Node representation matrix of $\widetilde{\mathcal{G}}^{h\text{-}th}$
$\mathbf{C}^{h\text{-}th} \in \mathbb{R}^{d \times N}$	*h-th* path attention weight matrix
$\mathbf{Z}_S \in \mathbb{R}^{N \times d}$	Structure-enhanced representation matrix
$\mathbf{Z}_F \in \mathbb{R}^{N \times d}$	Fused representation matrix
$\widehat{\mathbf{A}} \in \mathbb{R}^{N \times N}$	Rebuilt adjacency matrix
$\widehat{\mathbf{X}} \in \mathbb{R}^{N \times D}$	Rebuilt attribute matrix
$\cup, \cap, \varnothing$	Union symbol/intersection symbol/empty set

the information overlap among multi-order adjacency matrices, we further refine $\mathcal{E}_M$ through a two-step approach. Firstly, we rewrite (1) by removing the self-loop for each node:

$$\mathbf{A}^{h\text{-}th} = \mathbf{A}^{1st}\mathbf{A}^{(h-1)\text{-}th} - \operatorname{diag}(\mathbf{A}^{1st}\mathbf{A}^{(h-1)\text{-}th}), \qquad (2)$$

where $\operatorname{diag}(\mathbf{H})$ denotes a diagonal matrix of $\mathbf{H}$. For $\mathbf{A}^{h\text{-}th}$, we then drop the connections in $\mathcal{E}^{h\text{-}th}$ that overlap with the ones in $\mathcal{E}^{(h-1)\text{-}th}, \mathcal{E}^{(h-2)\text{-}th}, \ldots, \mathcal{E}^{1st}$. By doing this, the overlapping portion can be effectively eliminated, and $\mathcal{E}^{H\text{-}th} \cap \mathcal{E}^{(H-1)\text{-}th} \cap \cdots \cap \mathcal{E}^{1st} = \varnothing$. To enhance the local topology modeling capability of our model, we randomly mask the edges between two $h\text{-}th$-order connected attribute-missing nodes from $\mathcal{E}_M$ and denote a masked version of $\mathcal{E}_M$ as $\widetilde{\mathcal{E}}_M = \{\widetilde{\mathcal{E}}^{h\text{-}th}\}_{h=1}^{H}$. According to these mathematical formulations, edge-masked attribute-missing graphs with different orders can be formulated as $\widetilde{\mathcal{G}}^{1st} = \{\mathcal{V}^o, \mathcal{T}^m, \widetilde{\mathcal{E}}^{1st}\}, \widetilde{\mathcal{G}}^{2nd} = \{\mathcal{V}^o, \mathcal{T}^m, \widetilde{\mathcal{E}}^{2nd}\}, \ldots, \widetilde{\mathcal{G}}^{H\text{-}th} = \{\mathcal{V}^o, \mathcal{T}^m, \widetilde{\mathcal{E}}^{H\text{-}th}\}$, respectively.

Definition 3 (Learning task): This paper mainly addresses the attribute-missing problem on graphs without label annotation. To handle this problem, our model works for learning an encoder-decoder framework $\mathcal{F} = \mathcal{F}_d \circ \mathcal{F}_e$ to recover missing information from observations. The encoding function $\mathcal{F}_e(\cdot)$ accepts both $\widetilde{\mathcal{G}}$ and $\widetilde{\mathcal{G}}_M = \{\widetilde{\mathcal{G}}^{h\text{-}th}\}_{h=1}^{H}$, and produces the graph embedding $\mathbf{Z}_F \in \mathbb{R}^{N \times d}$ such that $d \ll D$, where we have conducted the unreliable information filtering and the local topology enhancing operations for attribute-missing samples in the latent space. According to the learned graph embedding, the decoding function $\mathcal{F}_d(\cdot)$ would recover the attribute-missing samples that can be saved and used for downstream tasks.

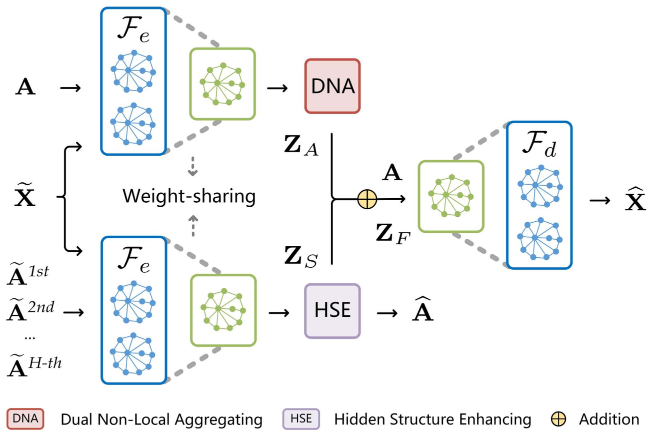
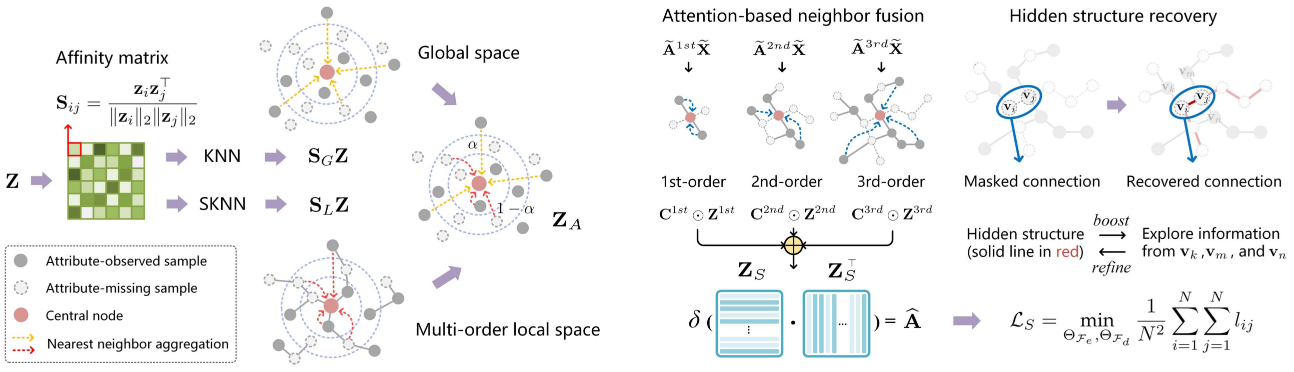


Fig. 1. The overall architecture of WAGE. Our weight-sharing architecture consists of two branches where the attribute and structure matrices are closely entangled. Specifically, in the upper branch, the DNA module improves the quality of missing attribute estimation by introducing an unreliable connection filtering mechanism. While in the bottom branch, the HSE module enables the network to focus more on the topology relationships of attribute-missing samples by having structure-attribute information verify each other.

B. Overview

This section will introduce and analyze our proposed WAGE framework in detail. Before our work, the primary solution for addressing the attribute-missing issues is first to encode observed attributes and graph structures independently, and then perform two-source information interaction in the latent space [11], [12], [18], [20]. In contrast, to establish a structure-attribute mutually enhanced learning strategy, our paper proposes a weight-sharing architecture where the attributes and the adjacency matrix are closely entangled. As illustrated in Fig. 1, the overall framework mainly consists of two branches. In the upper branch, the graph encoder $\mathcal{F}_e(\cdot)$ first accepts an initially imputed attribute matrix $\widetilde{\mathbf{X}}$ and a normalized adjacency matrix $\widetilde{\mathbf{A}}$ as inputs, then outputs an attribute-enhanced representation matrix $\mathbf{Z}_A \in \mathbb{R}^{N \times d}$ via dual non-local aggregating (DNA) module (Section III-C1 for details). The goal of the DNA design is to reveal unobserved high-confidence connections while filtering unreliable ones in the global and multi-order local spaces, as shown in Fig. 2(a). By this means, the network is enabled to discover more informative hints for accurate data imputation. While in the bottom branch, $\mathcal{F}_e(\cdot)$ once again accepts $\widetilde{\mathbf{X}}$ and multi-order edge-masked adjacency matrices $\widetilde{\mathbf{A}}^{1st}, \widetilde{\mathbf{A}}^{2nd}, \ldots, \widetilde{\mathbf{A}}^{H\text{-}th}$, and then outputs a structure-enhanced representation matrix $\mathbf{Z}_S \in \mathbb{R}^{N \times d}$ and a rebuilt adjacency matrix $\widehat{\mathbf{A}} \in \mathbb{R}^{N \times N}$ via hidden structure enhancing (HSE) module (Section III-C2 for details). Notably, HSE is considered an auxiliary task for missing attribute estimation. As shown in Fig. 2(b), by conducting attention-based neighbor fusion and hidden structure recovery, HSE enables the network to pay more attention to the topology reconstruction of attribute-missing samples, thereby assisting the DNA module in accurately estimating missing attributes. After that, the integration of $\mathbf{Z}_A$ and $\mathbf{Z}_S$ can further improve the quality of data imputation. Finally, the fused representation matrix $\mathbf{Z}_F$ is transferred into the decoder $\mathcal{F}_d(\cdot)$ for missing attribute reconstruction (Section III-D for details). In the following


Affinity matrix
$\mathbf{S}_{ij} = \frac{\mathbf{z}_i\mathbf{z}_j^\top}{\|\mathbf{z}_i\|_2\|\mathbf{z}_j\|_2}$
Z
KNN
$\mathbf{S}_G\mathbf{Z}$
SKNN
$\mathbf{S}_L\mathbf{Z}$
Global space
α
1 − α
$\mathbf{Z}_A$
Multi-order local space
Attribute-observed sample
Attribute-missing sample
Central node
Nearest neighbor aggregation
Attention-based neighbor fusion
$\tilde{\mathbf{A}}^{1st}\tilde{\mathbf{X}}$
$\tilde{\mathbf{A}}^{2nd}\tilde{\mathbf{X}}$
$\tilde{\mathbf{A}}^{3rd}\tilde{\mathbf{X}}$
1st-order
2nd-order
3rd-order
$\mathbf{C}^{1st}\odot\mathbf{Z}^{1st}$
$\mathbf{C}^{2nd}\odot\mathbf{Z}^{2nd}$
$\mathbf{C}^{3rd}\odot\mathbf{Z}^{3rd}$
$\mathbf{Z}_S$
$\mathbf{Z}_S^\top$
$\delta($ · $) = \hat{\mathbf{A}}$
Hidden structure recovery
Masked connection
Recovered connection
Hidden structure (solid line in red)
boost
refine
Explore information from $\mathbf{v}_k$, $\mathbf{v}_m$, and $\mathbf{v}_n$
$\mathcal{L}_S = \min_{\Theta_{\mathcal{F}_e},\Theta_{\mathcal{F}_d}} \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} l_{ij}$

(a) Dual Non-Local Aggregating (b) Hidden Structure Enhancing

Fig. 2. Illustration of the DNA module and the HSE module. In our work, we propose to solve the attribute-missing problem in a brand new perspective, *i.e.,* 1) the DNA module is designed to aggregate more informative features to complete the missing information in the global and multi-order local spaces. By filtering unreliable connections using indicator matrices (*i.e.,* $\mathbf{S}_G$ and $\mathbf{S}_L$), the DNA module can help the network discover more valuable clues for accurate data imputation; 2) the HSE module is designed to have two kinds of information verify each other by preserving multi-order information and conducting structure information recovery. As an auxiliary task, the HSE module guides the network to pay close attention to enhancing the topology relationships of attribute-missing samples for more accurate missing attribute estimation. By passing the information through the weight-sharing framework and combining two-source latent variables, the quality of data imputation could be improved.

sections, we will present a comprehensive illustration for each component.

C. Structure-Attribute Mutual Enhancement

In this section, we present the dual non-local aggregating (DNA) and hidden structure enhancing (HSE) modules. Both are illustrated in Fig. 2. The graph encoding processes will be discussed in detail as we explain the two modules.

1) Dual Non-Local Aggregating: For a given graph $\widetilde{\mathcal{G}}$, a weight-sharing graph encoder $\mathcal{F}_e(\cdot)$ conducts the following layer-wise propagation to compute l-th node representations $\mathbf{Z}^{(l)}$, of which i-th row, $\mathbf{z}_i^{(l)}$ denotes the representation vector for node $\mathbf{v}_i \in \mathcal{V}$:

$$\mathbf{Z}^{(l)} = \sigma(\mathbf{A}\mathbf{Z}^{(l-1)}\mathbf{W}^{(l)}), \tag{3}$$

where $\mathbf{W}^{(l)}$ denotes the learnable parameter matrix of the l-th encoder layer. σ is a non-linear activation function, *e.g.,* ReLU. Note that $\mathbf{Z}^{(0)}$ denotes the initially imputed attribute matrix $\widetilde{\mathbf{X}}$ of $\widetilde{\mathcal{G}}$. As observed in (3), the graph convolutional networks (GCNs)-based encoder performs neighbor aggregation at each layer, which can be interpreted as the process of partial neighbor reconstruction [54]. Hence, the network calculation process gradually imputes the missing attributes, leading to a complete representation matrix $\mathbf{Z}^{(2)}$, *i.e.,* $\mathbf{Z} \in \mathbb{R}^{N\times d}$. Although neighbor aggregation could be considered an initial data completion, relying solely on preserved neighbors in unsupervised scenarios may lead to biased and less discriminative latent features. This is because some initial low-confidence values stemmed from $\widetilde{\mathbf{X}}$ have diffused throughout the network during sample embedding, thus compromising the quality of the final imputed data. To overcome this issue, it is more intuitive to reveal unobserved highly correlated knowledge while filtering unreliable information for attribute-missing samples. Recent studies (*e.g.,* NLGCN [55] and CGAT [56]) have demonstrated the effectiveness of non-local aggregation for mining underlying valuable information. This motivates us to design the DNA module, which collects more informative non-local features to complete the missing information in the global and multi-order local spaces. By employing the K-nearest neighbor (KNN) search to filter unreliable connections, the DNA module helps the network discover more valuable clues for reliable data imputation. Specifically, we conduct an additional operation following $\mathcal{F}_e(\cdot)$:

$$\mathbf{Z}_A = \alpha\mathbf{S}_G\mathbf{Z} + (1-\alpha)\mathbf{S}_L\mathbf{Z}, \tag{4}$$

where α is a fusion coefficient that is set to 0.5 for initialization. $\mathbf{S}_G \in \mathbb{R}^{N\times N}$ and $\mathbf{S}_L \in \mathbb{R}^{N\times N}$ are global-scope and multi-order local-scope indicator matrices constructed in two different manners, respectively. As shown in Fig. 2, to construct $\mathbf{S}_G$ and $\mathbf{S}_L$, we first generate the affinity matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$ based on representation similarities of all samples:

$$\mathbf{S}_{ij} = \frac{\mathbf{z}_i\mathbf{z}_j^\top}{\|\mathbf{z}_i\|_2\|\mathbf{z}_j\|_2}, \quad \forall\ i,j \in [1,N], \tag{5}$$

where $\|\cdot\|_2$ denotes the ℓ_2-normalization. $\mathbf{z}_i$ and $\mathbf{z}_j$ refer to the representations of node $\mathbf{v}_i$ and node $\mathbf{v}_j$, respectively.

To improve the reliability of $\mathbf{S}$, two mechanisms are introduced. On the one hand, we search nearest neighbors for each node $\mathbf{v}_i$ from the global scope. The underlying idea is to discover non-adjacent nodes that share similar semantic features with the central node. For instance, in a citation network where each node indicates a specific paper and each edge indicates the citation relationship between two papers. Even though the research content of the two papers belongs to the same research field (*i.e.,* two nodes with similar features probably have the same label), they may not cite each other's work or share any citation in the graph. We argue that such semantically similar entities that do not share a connection can be exploited via K-nearest neighbor search from the global scope. To this end, we adopt a KNN strategy to filter the less confident similarity estimation in $\mathbf{S}$. Specifically, only the top K samples that are the most similar to each central sample are preserved, while others are discarded. To ensure consistency, we apply the Softmax function to normalize all similarities of top K samples. The resultant indicator matrix is represented as $\mathbf{S}_G$, where $\mathbf{S}_{Gij}$ is non-zero if the representation of node $\mathbf{v}_j$ is semantically similar to that of node $\mathbf{v}_i$ in the top K rankings, and $\mathbf{S}_{Gij}$ is zero otherwise. On the other hand, we conduct a structure-oriented K-nearest neighbor (SKNN) search to further boost the quality of the latent space from the multi-order neighbor scope. The intuition is that the multi-order neighbor nodes tend to share the same label as the central node. For example, in a citation network, even though two papers are not directly connected, they may be semantically similar and probably be in the same category since they have some common citations. Hence, it is worth exploring the informative features and filtering useless ones in the region of multi-order neighbor nodes. In this regard, to construct $\mathbf{S}_L$, we first preserve all similarities of $1st$- to $O\text{-}th$-order neighbors of each central sample and filter the non-neighbor ones according to $\mathbf{S}$. Similar to the constructing process of $\mathbf{S}_G$, we then conduct a KNN strategy to get the multi-order local-scope indicator matrix $\mathbf{S}_L$.

With $\mathbf{S}_G$ and $\mathbf{S}_L$, we employ two indicator matrices to update $\mathbf{Z}$ using (4), which enables reliable information aggregation for high-quality missing attribute estimation.

2) Hidden Structure Enhancing: Since graphs encompass both node attributes and graph structures, accurately recovering attribute-missing nodes requires a reflection of either the data characteristics or local topology within the graph. As a result, when attribute information is missing, it is important to rely more on the complete structure information to ensure the correctness of the learned representations, as the reconstruction of sample correlations usually reflects the quality of attribute embedding [57]. Motivated by this, we propose a hidden structure enhancing (HSE) module, which assists the DNA module in regularizing the latent space by preserving multi-order affinity information and recovering structure information. As an auxiliary task, the hidden structure enhancing process consists of two schemes, *i.e.,* attention-based neighbor fusion and hidden structure recovery, as illustrated in Fig. 2(b). Likewise, by transferring given graphs $\widetilde{\mathcal{G}}^{1st}, \widetilde{\mathcal{G}}^{2nd}, \ldots, \widetilde{\mathcal{G}}^{H\text{-}th}$ into the weight-sharing graph encoder $\mathcal{F}_e(\cdot)$, we can obtain the node representations of $\widetilde{\mathcal{G}}^{h\text{-}th}$:

$$\mathbf{Z}^{h\text{-}th(l)} = \sigma(\widetilde{\mathbf{A}}^{h\text{-}th}\mathbf{Z}^{h\text{-}th(l-1)}\mathbf{W}^{(l)}), \tag{6}$$

here, we empirically investigate the $1st$- to $3rd$-order neighbors within the graph data, where $h \in [1, 3]$. $\mathbf{Z}^{h\text{-}th(0)}$ and $\mathbf{Z}^{h\text{-}th(2)}$ refer to $\widetilde{\mathbf{X}}$ and the node representation matrix of $\widetilde{\mathcal{G}}^{h\text{-}th}$, respectively. Next, we transform these node representation matrices into a nonlinear transformation function (*e.g.,* one-layer multiple perception), so as to estimate the importance of $h\text{-}th$ path by calculating a normalized attention weight:

$$c_i^{h\text{-}th} = \frac{e^{\left(\mathbf{W}^{h\text{-}th}(\mathbf{z}_i^{h\text{-}th})^\top + \mathbf{b}^{h\text{-}th}\right)}}{\sum_{h=1}^{H} e^{\left(\mathbf{W}^{h\text{-}th}(\mathbf{z}_i^{h\text{-}th})^\top + \mathbf{b}^{h\text{-}th}\right)}}, \tag{7}$$

where $\mathbf{W}^{h\text{-}th}$ and $\mathbf{b}^{h\text{-}th}$ denotes the learnable parameter matrix of the attention layer and the bias vector in the $h\text{-}th$ path, respectively. A higher value of $c_i^{h\text{-}th}$ indicates that the $h\text{-}th$-order observed neighbors of node $\mathbf{v}_i$ can offer more valuable information. Subsequently, we integrate these representation matrices using attention weights:

$$\mathbf{Z}_S = \sum_{h=1}^{H} \mathbf{C}^{h\text{-}th} \odot \mathbf{Z}^{h\text{-}th}, \tag{8}$$

where $\odot$ means Hadamard product. $\mathbf{C}^{h\text{-}th} \in \mathbb{R}^{N \times d}$ represents the attention matrix, and $\mathbf{c}_i^{h\text{-}th} \in \mathbb{R}^d$ is the attention vector of node $\mathbf{v}_i$, which repeats $c_i^{h\text{-}th}$ with d times. After that, we can obtain the rebuilt adjacency matrix $\widehat{\mathbf{A}} \in \mathbb{R}^{N \times N}$ by introducing a Sigmoid function $\mathrm{Sigm}(\cdot)$ to conducting matmul product over $\mathbf{Z}_S$:

$$\widehat{\mathbf{A}} = \mathrm{Sigm}(\mathbf{Z}_{\mathrm{S}}\mathbf{Z}_{\mathrm{S}}^\top). \tag{9}$$

According to (9), we can model the sample correlation between node $\mathbf{v}_i$ and node $\mathbf{v}_j$ by minimizing the error:

$$e_{ij} = -\left(\mathbf{A}_{ij} \ln \widehat{\mathbf{A}}_{ij} + (1 - \mathbf{A}_{ij}) \ln(1 - \widehat{\mathbf{A}}_{ij})\right), \tag{10}$$

where there exists a linkage between node $\mathbf{v}_i$ and node $\mathbf{v}_j$ in the original graph if $\mathbf{A}_{ij}$ is non-zero, otherwise $\mathbf{A}_{ij}$ is zero.

Finally, we introduce a pre-defined hyper-parameter γ into the edge reconstruction process. This parameter is utilized to balance the two types of edge reconstruction processes, *i.e.,* masked edge reconstruction and unmasked edge reconstruction, which correspond to the dotted and solid gray lines in Fig. 2(b):

$$l_{ij} = \begin{cases} \gamma e_{ij}, & \mathbf{v}_i, \mathbf{v}_j \in \mathcal{V}^m \\ e_{ij}, & \text{otherwise} \end{cases}. \tag{11}$$

Our method differs from existing works [11], [12], [18], [20] in terms of structure reconstruction fashion. Unlike previous works that treat all edges equally, HSE guides the network to focus more on preserving the correct topology relationships of attribute-missing samples through a weighted edge reconstruction scheme. As a result, WAGE is encouraged to facilitate a deeper understanding of the relationships between reconstructed attribute-missing samples in the graph, which in turn discovers more valuable signals from high-order neighbor samples for data completion. The final structure reconstruction loss of WAGE can be formulated as:

$$\mathcal{L}_S = \min_{\Theta_{\mathcal{F}_e}, \Theta_{\mathcal{F}_d}} \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} l_{ij}. \tag{12}$$

D. Decoding and Training Objectives

1) Decoding: A good graph imputation model should consider the inherent properties of attribute-missing samples in terms of both data characteristics and topology. On the one hand, the DNA module encourages high-quality data imputation by promoting the retention of important features while eliminating extraneous ones. However, it is unclear whether this component can accurately preserve topology relationships of reconstructed attribute-missing samples within a graph. On the other hand, the HSE module serves as an auxiliary task to provide the DNA module with more reliable structure features by having structure-attribute information verify each other. To

Algorithm 1: Pre-training Procedure for WAGE.

```
Input: Inputs $\widetilde{\mathcal{G}}, \widetilde{\mathcal{G}}^{1st}, \widetilde{\mathcal{G}}^{2nd}, \ldots, \widetilde{\mathcal{G}}^{H\text{-}th}$; maximum
  iteration number $I$; number of neighbor orders $O$; number
  of nearest neighbors $K$; balanced coefficient $\gamma$, $\lambda$;
  learning rate $\eta$; model parameters $\{\Theta_{\mathcal{F}_e}, \Theta_{\mathcal{F}_d}\}$.
Output: Pre-trained model parameters $\{\Theta_{\mathcal{F}_e}, \Theta_{\mathcal{F}_d}\}$.
Initialize $\{\Theta_{\mathcal{F}_e}, \Theta_{\mathcal{F}_d}\}$ with an Xavier manner;
for $i = 1$ to $I$ do
    $\mathbf{Z} \leftarrow$ Obtain representations with $\mathcal{F}_e(\cdot)$;
    $\mathbf{S} \leftarrow$ Construct the affinity matrix by (5);
    $\mathbf{S}_G, \mathbf{S}_L \leftarrow$ Conduct KNN and SKNN based on $\mathbf{S}$;
    $\mathbf{Z}_A \leftarrow$ Obtain representations by (4);
    $\mathbf{Z}^{1st}, \mathbf{Z}^{2nd}, \mathbf{Z}^{3rd} \leftarrow$ Obtain representations with
      $\mathcal{F}_e(\cdot)$;
    $\mathbf{Z}_S \leftarrow$ Obtain representations by (7) and (8);
    $\widehat{\mathbf{A}} \leftarrow$ Reconstruct adjacency matrix by (9);
    $\mathcal{L}_S \leftarrow$ Calculate loss by (10)–(12);
    $\mathbf{Z}_F \leftarrow$ Integrate $\mathbf{Z}_A$ and $\mathbf{Z}_S$ by (13);
    $\widehat{\mathbf{X}} \leftarrow$ Output raw attribute from $\mathbf{Z}_F$ with $\mathcal{F}_d(\cdot)$;
    $\mathcal{L}_A \leftarrow$ Calculate loss by (15);
    $\mathcal{L} \leftarrow$ Calculate the total loss by (16).
    Update $\{\Theta_{\mathcal{F}_e}, \Theta_{\mathcal{F}_d}\}$ by calculating:
      $\Theta_{\mathcal{F}_e} \leftarrow \Theta_{\mathcal{F}_e} - \eta\nabla_{\Theta_{\mathcal{F}_e}}\mathcal{L}$;
      $\Theta_{\mathcal{F}_d} \leftarrow \Theta_{\mathcal{F}_d} - \eta\nabla_{\Theta_{\mathcal{F}_d}}\mathcal{L}$;
end for
return $\Theta_{\mathcal{F}_e}, \Theta_{\mathcal{F}_d}$
```

achieve more comprehensive data completion, we combine the attribute-enhanced representation matrix $\mathbf{Z}_A$ and the structure-enhanced representation matrix $\mathbf{Z}_S$ from both modules via a fusion coefficient β that is initialized as 0.5. Then we directly feed the fused representation matrix $\mathbf{Z}_F$ and $\mathbf{A}$ into a graph decoder $\mathcal{F}_d(\cdot)$, which can be formulated as:

$$\mathbf{Z}_F = \beta\mathbf{Z}_A + (1-\beta)\mathbf{Z}_S, \tag{13}$$

$$\widetilde{\mathbf{Z}}^{(l)} = \sigma(\mathbf{A}\widetilde{\mathbf{Z}}^{(l-1)}\widetilde{\mathbf{W}}^{(l)}), \tag{14}$$

where $\widetilde{\mathbf{W}}^{(l)}$ denotes the learnable parameter matrix of the l-th decoder layer. $\widetilde{\mathbf{Z}}^{(0)}$ and $\widetilde{\mathbf{Z}}^{(2)}$ refer to $\mathbf{Z}_F$ and the rebuilt attribute matrix $\widehat{\mathbf{X}}$, respectively.

2) Training Objectives: The training objective of WAGE consists of two parts, *i.e.,* the observed attribute reconstruction loss, and the structure reconstruction loss associated with the HSE module:

$$\mathcal{L}_A = \min_{\Theta_{\mathcal{F}_e},\Theta_{\mathcal{F}_d}} \frac{1}{2N^o}\|\widetilde{\mathbf{X}}^o - \widehat{\mathbf{X}}^o\|_F^2. \tag{15}$$

$$\mathcal{L} = \min_{\Theta_{\mathcal{F}_e},\Theta_{\mathcal{F}_d}} \lambda\mathcal{L}_A + \mathcal{L}_S. \tag{16}$$

In (16), $\mathcal{L}_A$ denotes the mean square error (MSE) between the observed parts of $\widetilde{\mathbf{X}}$ and $\widehat{\mathbf{X}}$. λ is a pre-defined hyper-parameter that balances the importance of both reconstruction processes. The detailed learning procedure of the proposed WAGE is shown in Algorithm 1. Compared to existing attribute-missing graph learning methods, we design a different and effective framework to solve the newly proposed problem, *i.e.,* unsupervised graph learning on attribute-missing graphs. Here, we summarize the merits of our WAGE as below: 1) more comprehensive data completion. To complete attribute-missing samples, we take the proposed DNA and HSE modules to achieve reliable attribute imputation and effective topology reconstruction for attribute-missing samples, respectively; 2) a more compact data-completing architecture. Our method differs from ITR [12] and CAST [18] in that WAGE entangles the learning of attribute embedding and structure embedding through a weight-sharing encoder, enabling us to take full advantage of the two types of information. Notably, WAGE is a single autoencoder design without incorporating additional parameterized encoders or decoders; and 3) free of overly strict assumptions. Instead of adopting a structure-attribute distribution alignment approach [11], [20], we fill up the missing values by reconstructing them with the visible values in the reliable samples that are linked by the complete adjacent values.

IV. Experiments

In this section, we evaluate the benefits of WAGE against advanced graph learning methods. The experiments aim to answer the following six research questions:

- **RQ1**. Does WAGE perform better than baseline methods in downstream tasks? (see Section IV-B)
- **RQ2**. How does imputation initialization influence the performance of WAGE? (see Section IV-C)
- **RQ3**. How does each proposed component influence the performance of WAGE? (see Section IV-D)
- **RQ4**. Can WAGE still work better than competitors with less observed information? (see Section IV-E)
- **RQ5**. How do key hyper-parameters influence the performance of WAGE? (see Section IV-F)
- **RQ6**. How about the model convergence and performance stability? (see Section IV-G)

In the following, we begin with a brief introduction of the experimental setup and then report experiment results with detailed analyses.

A. Experimental Setup

1) Benchmark Datasets: We conduct experiments on six benchmark datasets, including four datasets with categorial attributes, *i.e.,* Cora [58], Citeseer [59], Amazon Photo (Amap for abbreviation), and Amazon Computer (Amac for abbreviation) [60], and two datasets with real-valued attributes, *i.e.,* Pubmed [59] and Coauthor-CS (CoCS for abbreviation) [61]. Table II summarizes the brief information of these datasets.

- Cora, Citeseer, and Pubmed are three popular citation network datasets. Specially, nodes mean scientific publications, and edges mean citation relationships. Each node has a predefined feature with corresponding dimensions.
- Amac and Amap are segments of the Amazon co-purchase network, where nodes represent goods, edges indicate that two goods are frequently bought together, node features are

TABLE II
SUMMARY OF BENCHMARK DATASETS

Dataset	Nodes	Edges	Dimension	Classes
Cora	2,708	5,278	1,433	7
Citeseer	3,327	4,228	3,703	6
Amac	13,752	245,861	767	10
Amap	7,650	119,081	745	8
Pubmed	19,717	44,324	500	3
CoCS	18,333	81,894	6,805	15

bag-of-words encoded product reviews, and class labels are given by the product category.

- CoCS is a Microsoft Academic Graph, where nodes mean authors and edges mean connections that connect two authors if they co-author a paper.

2) Training Procedure: The training procedure of WAGE includes two steps in total. Firstly, in the pre-training step, all available graph information is fed into WAGE for at least 300 training iterations until convergence by minimizing (16). During the training process, we utilize the weighted Binary Cross-Entropy loss (BCE) and the Mean Square Error (MSE) loss as training objectives for categorical and real-valued graph data, respectively, as was done in SAT [11]. To avoid the over-fitting issue, we adopt an early stop strategy when the loss value comes to a plateau. Secondly, in the node classification task, we utilize a graph convolutional network to perform node classification over the reconstructed attribute-missing nodes. The classifier is trained for 1000 iterations using a cross-entropy loss function, and we employ five-fold validation repeated ten times to ensure accuracy. The accuracy (ACC) metric is used to evaluate the average performance of the classifier. It is worth noting that we strictly adhere to the classifier setting in SAT [11] for node classification. Additionally, in the profiling task, we measure the recall and ranking quality of the reconstructed attribute-missing nodes using Recall and NDCG as metrics.

3) Implementation Details: We conduct experiments on six benchmark datasets using the official source code of all compared methods, except for Amer [20], and report the reproduced performance. For Amer, we record the results directly according to the original papers due to the unavailable or inaccessible source code. For the proposed WAGE, 1) during the pre-training stage, we randomly select 40% of the nodes with attributes as the training set. We then mask all attributes of the remaining 10% and 50% of nodes, which we refer to as attribute-missing nodes, to form the validation set and test set, respectively. We pre-train our WAGE (*i.e.,* a four GCN-layer autoencoder framework) using Adam optimizer with a learning rate set to 1e-3. According to the results of parameter sensitivity testing, we set the number of neighbor orders O and the number of nearest neighbors K as 5, and meanwhile, fix two balanced coefficients γ and λ to 5 and 10, respectively; 2) in the downstream tasks, the reconstructed attribute-missing nodes are then randomly split into 80% train data and 20% test data. Moreover, the learning rate, the latent dimension, the dropout rate, and the weight decay are set to 1e-3, 64, 0.5, and 5e-4, respectively. Note that these hyper-parameters are not carefully tuned for ease of model learning. In our settings, the used classifier receives $\widehat{\mathbf{X}}^m \in \mathbb{R}^{N^m \times D}$ and $\widetilde{\mathbf{A}}^m \in \mathbb{R}^{N^m \times N^m}$ as inputs, where $\widehat{\mathbf{X}}^m$ and $\widetilde{\mathbf{A}}^m$ denote the reconstructed attribute matrix of $\mathcal{V}^m$ and the corresponding adjacency matrix. To ensure a fair and accurate comparison, the reported results of WAGE and compared baselines were conducted on the same device (*e.g.,* Linux servers with NVIDIA 3090 GPU cards) and under identical configuration settings (*e.g.,* PyTorch version 1.12.0, DGL version 0.9.0, CUDA version 11.3, Numpy version 1.22.3, Scikit-learn version 1.1.1, and Networkx version 2.8.4). Furthermore, we strictly adhere to the same data splits as those employed in SAT [11] for all benchmark datasets, including the split of attribute-observed/-missing nodes and the spilt of train/test sets.

4) Baseline Methods: In this work, we compare our WAGE with eighteen published baseline methods to illustrate its effectiveness and superiority. Based on the completeness of data attributes, we can classify these baseline methods into three categories:

- Attribute-complete methods. **NeighAggre** [62] is a classical profiling method. **GCN** [28], **GraphSage** [63], and **GAT** [64] are three typical graph neural networks (GNNs). **Hers** [65] is one representative cold-start recommendation method. **GraphRNA** [66] and **ARWMF** [67] are two typically attributed random walk-based methods. **VAE** [68] and **GraphMAE** [53] are two representative autoencoder-based generative methods.
- Attribute-incomplete methods. **GINN** [9], **GCNMF** [10], **FP** [48], and **T2-GNN** [49] are four state-of-the-art attribute-incomplete graph learning methods.
- Attribute-missing methods. **ITR** [12], **SAT-GCN** [11], **SAT-GraphSage** [11], **SAT-GAT** [11], and **Amer** [20] are five advanced attribute-missing graph learning methods.

Note that in our problem, we test all aforementioned baseline methods on attribute-missing graphs and report their performance for comparison.

B. Overall Performance (RQ1)

1) Node Classification Task: As reported in Table III, we report the node classification performance of eighteen compared methods on six benchmark datasets. From these results, some key observations can be obtained: 1) the proposed WAGE achieves the best average accuracy performance on all datasets, with margins going up to 0.96% - 48.71%. These benefits demonstrate the effectiveness and superiority of WAGE for handling attribute-missing graphs in unsupervised scenarios; 2) GCN, GraphSage, and GAT are three well-known node classification networks. However, when dealing with attribute-missing samples, these methods suffer from a significant decrease in performance compared to our proposed method. This is due to the fact that they do not possess a specialized feature completion mechanism for attribute-missing samples, which limits their ability to learn effective representations from absent graph data; 3) GraphMAE is considered one of the strongest attribute-complete graph learning methods, while its performance is not comparable to ours. The reason behind this is that GraphMAE may fail to fully capture the intricate patterns within the graph when attributes are missing. As a result, its ability to generalize from available data to the masked content

TABLE III
NODE CLASSIFICATION PERFORMANCE COMPARISON ON SIX BENCHMARK DATASETS

Type	Method	Cora	Citeseer	Amac	Amap	Pubmed	CoCS	Avg. ↓
AC	NeighAggre (PNAS '08)	64.76	54.19	87.13	90.10	65.70	80.31	10.60 ↓
	GCN (ICLR '17)	44.27	40.52	40.19	37.77	42.08	23.33	46.26 ↓
	GraphSage (NeurIPS '17)	60.13	43.67	40.13	38.26	42.19	28.20	42.19 ↓
	GAT (ICLR '18)	45.95	27.09	40.15	37.80	42.08	23.32	48.22 ↓
	Hers (AAAI '19)	34.64	32.87	40.99	38.44	41.75	24.77	48.71 ↓
	GraphRNA (KDD '19)	81.33	64.02	86.89	90.13	**81.76**	87.32	2.38 ↓
	ARWMF (NeurIPSW '19)	80.65	27.23	73.88	61.78	81.45	83.78	16.17 ↓
	VAE (ICLR '14)	29.29	26.69	44.38	52.45	40.09	21.80	48.51 ↓
	GraphMAE (KDD '22)	75.06	67.89	84.37	88.26	77.14	82.15	5.14 ↓
AI	GINN (NN '20)	67.58	55.32	81.72	87.77	54.28	79.74	13.22 ↓
	GCNMF (FGCS '21)	70.30	63.40	76.43	87.79	62.00	87.53	9.71 ↓
	FP (LOG '22)	59.58	39.72	43.15	46.11	50.78	OOM	35.49 ↓
	T2-GNN (AAAI '23)	74.49	67.33	OOM	71.29	OOM	OOM	11.51 ↓
AM	ITR (IJCAI '22)	85.16	68.11	87.72	91.53	OOM	OOM	0.96 ↓
	SAT-GCN (T-PAMI '22)	83.24	66.04	84.98	89.12	74.21	84.51	3.94 ↓
	SAT-GraphSage (T-PAMI '22)	82.74	65.70	88.18	92.00	73.87	80.54	3.78 ↓
	SAT-GAT (T-PAMI '22)	85.35	67.32	87.60	**92.50**	76.55	84.19	2.04 ↓
	Amer† (T-CYB '22)	80.21	66.95	79.06	90.46	77.52	**89.22**	3.72 ↓
	WAGE (ours)	**85.90**	**69.33**	**88.67**	92.40	80.50	88.92	

† The results (if available) are recorded from the original paper directly, as the source code is inaccessible.

"AC", "AI", and "AM" refer to the attribute-complete methods, the attribute-incomplete methods, and the attribute-missing methods, respectively. "OOM" means out-of-memory error. "Avg. ↓" indicates the average degradation in performance across all datasets when compared to WAGE. The **boldface** value indicates the best results.

TABLE IV
PROFILING PERFORMANCE COMPARISON ON CORA AND CITESEER

Method	Cora						Citeseer					
	Recall@10	Recall@20	Recall@50	NDCG@10	NDCG@20	NDCG@50	Recall@10	Recall@20	Recall@50	NDCG@10	NDCG@20	NDCG@50
NeighAggre	9.06	14.13	19.61	12.17	15.48	18.50	5.11	9.08	15.01	8.23	11.55	15.60
GCN	12.56	17.85	29.73	17.21	20.76	27.04	6.28	10.97	20.49	10.31	14.21	20.44
GraphSage	12.91	18.10	30.24	17.97	21.45	27.86	5.60	10.63	19.90	9.78	13.56	19.97
GAT	12.67	17.93	29.70	17.30	20.87	27.10	5.62	10.12	19.56	8.79	12.53	18.71
VAE	8.87	12.33	20.97	12.23	14.56	19.16	3.82	6.69	12.94	6.01	8.40	12.50
GraphMAE	3.69	4.39	7.48	5.74	7.25	10.79	1.27	3.54	5.35	2.23	4.91	7.51
GraphRNA	14.05	20.83	31.62	19.39	23.69	29.42	7.79	12.83	22.65	13.01	17.24	23.44
ARWMF	12.99	18.03	29.80	18.64	22.14	27.96	5.56	10.18	19.59	8.46	12.25	18.29
Hers	12.06	17.08	27.87	17.03	20.52	25.56	5.74	10.22	19.70	9.00	12.85	19.11
GINN	13.12	18.67	28.87	18.28	21.66	27.78	6.07	10.53	20.22	9.31	13.46	19.89
GCNMF	13.43	18.99	29.05	18.78	22.04	28.08	6.35	11.01	20.75	9.84	13.87	20.01
FP	9.87	11.31	19.02	11.23	13.18	19.98	4.02	7.32	16.54	7.57	10.26	10.08
T2-GNN	12.26	15.35	22.15	17.20	19.28	22.81	5.27	8.59	15.54	9.46	12.24	16.79
ITR	16.80	23.67	36.01	23.20	27.51	34.65	9.55	15.06	26.32	16.28	20.91	28.38
SAT-GCN	14.75	21.30	33.24	20.71	24.98	31.71	7.55	12.61	23.38	13.05	17.26	24.33
SAT-GraphSage	14.48	20.22	31.89	20.01	23.92	30.61	7.45	11.82	21.94	12.23	16.47	22.75
SAT-GAT	16.36	23.47	35.95	22.62	27.32	33.99	8.02	13.37	24.40	13.79	18.24	25.48
WAGE (ours)	**17.28**	**24.10**	**36.51**	**23.84**	**28.45**	**35.01**	**9.68**	**15.39**	**26.75**	**16.55**	**21.24**	**28.68**

The **boldface** value indicates the best results.

may be compromised, leading to unsatisfying performance; 4) our method outperforms GraphRNA and ARWME on five of six benchmark datasets. Since these attributed random walk-based methods have the potential to capture the correlation between attribute dimensions [11], GraphRNA and ARWME excel at handling the real-valued graph data to some extent, thereby showing the most competitive performance compared to ours on Pubmed; 5) compared with GINN, GCNMF, FP, and T2-GNN, WAGE achieves an average accuracy increment of 13.22%, 9.71%, 35.49%, and 11.51%, respectively. The results suggest that these attribute-incomplete graph learning methods are limited in their abilities to achieve high-quality data imputation when node attributes are entirely missing, as the techniques designed for recovering attribute-incomplete samples may not be effective in attribute-missing scenarios; and 6) WAGE achieves average performance gain up to 0.96%-3.94% over five attribute-missing graph learning methods on all datasets. These improvements are attributed to the novel idea of establishing a structure-attribute mutually enhanced learning strategy for data completion.

2) *Profiling Task:* Tables IV and V summarize the profiling performance of seventeen methods on four datasets. The results reveal several major observations that are similar to those obtained from the node classification task: 1) WAGE exhibits superior performance compared to all other baseline methods across all datasets in terms of six metrics, demonstrating its effectiveness and superiority in processing attribute-missing graphs; 2) WAGE exhibits remarkably strong performance in comparison to two autoencoders, *i.e.,* VAE and GraphMAE, indicating that our autoencoder-style framework can be a promising alternative for learning node representations in the presence of missing attributes; 3) WAGE consistently outperforms the attributed rand walk-based methods. This is due to the fact

TABLE V
PROFILING PERFORMANCE COMPARISON ON AMAC AND AMAP

Method	Amac						Amap					
	Recall@10	Recall@20	Recall@50	NDCG@10	NDCG@20	NDCG@50	Recall@10	Recall@20	Recall@50	NDCG@10	NDCG@20	NDCG@50
NeighAggre	3.21	5.93	13.06	7.88	11.56	19.23	3.30	6.16	13.61	8.13	11.96	19.98
GCN	2.73	5.29	12.76	6.75	10.28	18.28	2.95	5.67	13.23	7.09	10.80	18.94
GraphSage	2.67	5.25	12.70	6.63	10.15	18.15	2.93	5.66	13.26	7.11	10.83	19.00
GAT	2.71	5.29	12.76	6.73	10.29	18.28	2.93	5.66	13.23	7.09	10.81	18.95
VAE	2.56	5.06	12.14	6.36	9.78	17.44	2.77	5.30	12.72	6.78	10.25	18.23
GraphMAE	1.30	2.60	6.51	4.57	7.09	12.99	1.34	2.68	6.71	4.68	7.25	13.29
GraphRNA	3.89	6.92	14.74	9.38	13.38	21.64	3.92	7.00	15.01	9.65	13.83	22.39
ARWMF	2.82	5.40	12.81	6.91	10.48	18.32	2.99	5.71	13.25	7.24	10.89	19.09
Hers	2.81	5.29	12.68	6.72	10.30	18.21	2.97	5.81	13.32	7.09	10.96	19.11
GINN	2.86	5.24	12.31	6.54	10.12	18.98	3.01	5.82	13.20	7.13	10.87	19.21
GCNMF	2.77	5.21	12.76	6.75	10.44	18.83	2.96	5.74	12.98	6.96	10.20	18.87
FP	2.41	5.07	11.89	6.47	9.65	17.89	2.93	5.04	12.24	6.97	10.33	19.14
T2-GNN	OOM	OOM	OOM	OOM	OOM	OOM	2.59	4.92	11.98	6.51	9.73	17.31
ITR	4.45	7.76	16.22	10.90	15.32	24.26	4.37	7.74	16.30	10.66	15.21	24.29
SAT-GCN	3.95	7.07	15.19	9.70	14.26	23.19	4.02	7.32	15.80	9.86	14.25	23.28
SAT-GraphSage	4.20	7.41	15.70	10.31	14.70	23.45	4.42	7.82	16.10	10.76	14.92	24.15
SAT-GAT	4.22	7.50	15.91	10.34	14.81	23.93	4.21	7.60	16.25	10.33	14.85	24.04
WAGE (ours)	**4.47**	**7.99**	**16.61**	**11.08**	**15.62**	**24.71**	**4.56**	**7.96**	**16.55**	**11.10**	**15.63**	**24.72**

"OOM" means out-of-memory error. The **boldface** value indicates the best results.

that random walks can introduce noise to the learning process, thereby impacting the quality of the reconstructed missing attributes; 4) WAGE outperforms GCN, GraphSage, and GAT by a significant margin, despite their proven ability to learn strong representations on complete graphs. This is because these methods are not well-suited for effectively handling graphs with missing attributes; 5) WAGE achieves an approximate 2.73%, 2.62%, 4.19%, and 3.80% average performance gain in terms of Recall@10 over GINN, GCNMF, FP, and T2-GNN on Cora, Citeseer, Amac, and Amap, respectively. This indicates that these attribute-incomplete graph learning methods fall into inaccurate data imputation with extremely limited observations so that they can not learn effective representations; and 6) with its stronger representation learning capability, the graph attention network enables SAT-GAT to consistently outperform its GCN-based and GraphSage-based counterparts. In addition, our method further outperforms SAT-GAT by 1.22%, 2.76%, 0.74%, and 0.77% performance improvements in terms of NDCG@10 metric across four datasets, which is attributed to both the novel architecture and new data completion strategy.

C. Influence of Initial Imputation (RQ2)

To ease the model learning, it is imperative for a given attribute-missing graph to assign initial values to attribute-missing nodes before network training. To illustrate the impact of different measures for initial imputation, we consider four scenarios and evaluate the model performance as reported in Table VI. "WAGE-G", "WAGE-N", "WAGE-T", and "WAGE-Z" are methods where the missing attribute vectors in $\mathbf{X}$ are filled with random Gaussian noise, observed neighbors, zero values, and token values. From the results in this table, we can observe that 1) "WAGE-G" is not comparable to "WAGE-N" and "WAGE-Z" on Citesser, Amac, Pubmed, and CoCS. The reason behind this is two-fold. On the one hand, random noise contains a large amount of semantically irrelevant information that has diffused throughout the network, adversely affecting the discriminative ability of the imputed features and even distorting the graph. On the other hand, the Gaussian distribution may not ideally conform to the graph data in some cases; 2) "WAGE-T" achieves poor performance. This is because tokens consist of a set of stochastic learnable vectors, which could potentially result in inaccurate interpretations or predictions in attribute-missing cases; and 3) "WAGE-N" and "WAGE-Z" exhibit comparable performance across six benchmark datasets, indicating that either selecting observed neighbors or assigning zero values for initializing missing attributes is equally effective.

TABLE VI
ABLATION STUDY ON THE FASHION OF INITIAL IMPUTATION

Method	Cora	Citeseer	Amac	Amap	Pubmed	CoCS
WAGE-G	85.11	67.23	87.65	90.45	74.33	84.71
WAGE-N	85.04	69.23	88.31	91.89	**81.30**	88.50
WAGE-T	68.59	62.33	50.97	90.00	42.79	OOM
WAGE-Z	**85.90**	**69.33**	**88.67**	**92.40**	80.50	**88.92**

"WAGE-G", "WAGE-N", "WAGE-T", and "WAGE-Z" are methods where the missing attribute vectors in $\mathbf{X}$ are filled with random Gaussian noise, observed neighbors, zero values, and token values. The **boldface** value indicates the best result.

D. Ablation Study on DNA and HSE (RQ3)

1) Effectiveness of Each Component: To demonstrate the effectiveness of each component, we compare WAGE with its four variants on four datasets. Concretely, the baseline method utilizes a pure graph autoencoder without incorporating any proposed module. "w/ HSE" indicates the WAGE method only utilizing the hidden structure enhancing module, while "w/ DNA" refers to the WAGE method only utilizing the dual non-local aggregating module. "w/o WS" denotes the WAGE method without using the weight-sharing backbone for information encoding. As shown in Fig. 3, some major observations can be summarized: 1) when compared to the baseline method, "w/ HSE" and "w/ DNA" produce performance gains of 0.92%/1.17%, 1.18%/1.84%, 0.82%/0.93%, and 0.68%/0.75% in terms of Recall@10 across four datasets, indicating that either the DNA module or the HSE module plays an important role in effectively handling attribute-missing graphs; 2) WAGE consistently improve the performance of "w/ HSE" and "w/

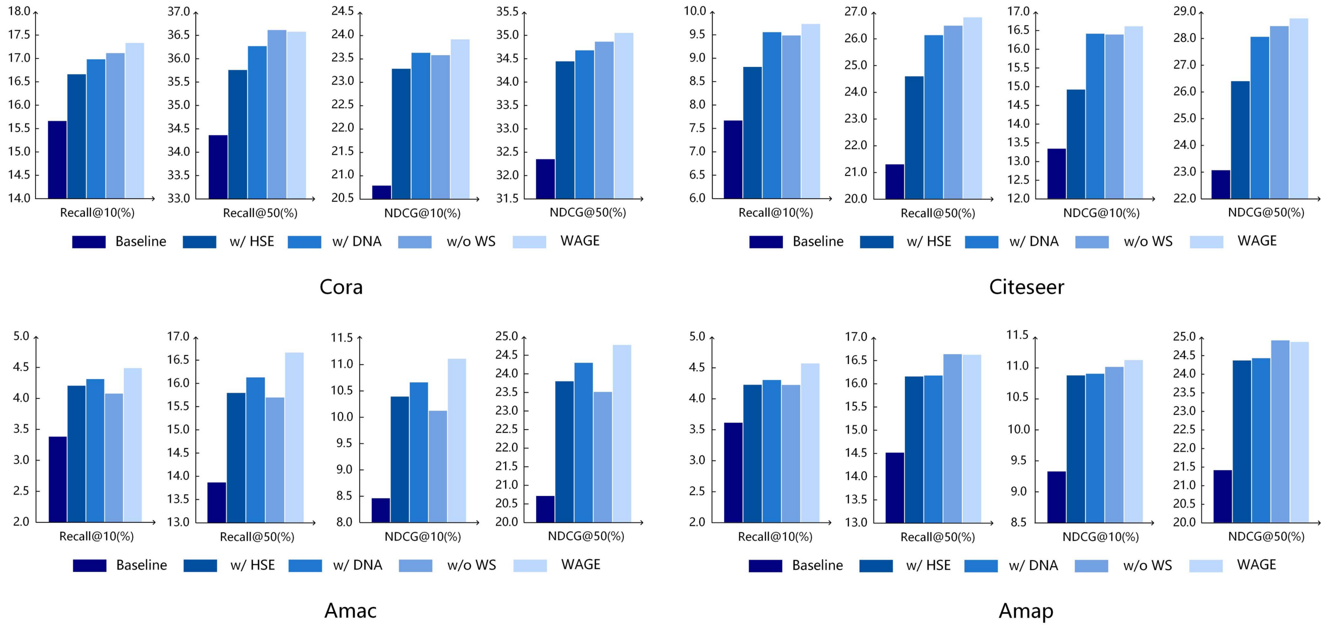


Fig. 3. Ablation study on DNA and HSE. Recall and NDCG using the top 10 and top 50 on four datasets are reported. The baseline method utilizes a pure graph autoencoder without incorporating any proposed module. “w/ HSE” indicates the WAGE method only utilizing the hidden structure enhancing module, while “w/ DNA” refers to the WAGE method only utilizing the dual non-local aggregating module. “w/o WS” denotes the WAGE method without using the weight-sharing backbone for information encoding. The performance of our method is listed in the last bar.

DNA” in terms of two metrics over all datasets. Taking the results on Citeseer for example, WAGE gains 2.14% and 0.65% Recall@50 performance increments over its two counterparts. These results indicate that a comprehensive approach that takes into account both data characteristics and topology is more effective in achieving accurate data completion than a singular focus on either aspect. Similar observations can be obtained from the results on other datasets; and 3) in most cases, WAGE demonstrates better performance than its “w/o WS” counterpart, which can be attributed to the fact that weight-sharing design can generalize across sub-tasks by capturing diverse information from different inputs for data imputation.

2) Influence of KNN and SKNN for DNA: To further reveal the effect of the DNA module, we compare WAGE with its two variants and report their Recall and NDCG performance on Cora, Citeseer, Amac, and Amap in Table VII. “w/o KNN” and “w/o SKNN” denote two WAGE variants with the K-nearest neighbor operation and the structure-oriented K-nearest neighbor operation being removed, respectively. From the results reported in the table, we can observe that 1) both “w/o KNN” and “w/o SKNN” achieve promising performance on Cora, which demonstrates the effectiveness of promoting the retention of important features while eliminating extraneous information for data imputation. Similar observations can be concluded from the results on other datasets; and 2) WAGE consistently enhances the performance of “w/o KNN” and “w/o SKNN” by uncovering useful clues in both global and multi-order local spaces. It is worth noting that although there is some overlap between these two types of information retrieval, empirical results demonstrate that combining them yields mutual benefits in exploiting more informative information to some extent.

TABLE VII
Ablation Study on the DNA Module

Method	Cora			
	Recall@10	Recall@50	NDCG@10	NDCG@50
w/o KNN	16.59	35.78	23.14	34.20
w/o SKNN	16.78	35.80	23.30	34.42
WAGE	**17.28**	**36.51**	**23.84**	**35.01**
Method	Citeseer			
	Recall@10	Recall@50	NDCG@10	NDCG@50
w/o KNN	9.24	26.46	15.77	28.10
w/o SKNN	9.31	26.32	16.18	28.25
WAGE	**9.68**	**26.75**	**16.55**	**28.68**
Method	Amac			
	Recall@10	Recall@50	NDCG@10	NDCG@50
w/o KNN	4.20	16.12	10.33	24.26
w/o SKNN	4.29	16.24	10.46	24.42
WAGE	**4.47**	**16.61**	**11.08**	**24.71**
Method	Amap			
	Recall@10	Recall@50	NDCG@10	NDCG@50
w/o KNN	4.31	16.30	10.68	24.38
w/o SKNN	4.37	16.08	10.74	24.30
WAGE	**4.56**	**16.55**	**11.10**	**24.72**

Recall and NDCG using the top 10 and top 50 on four datasets are reported. “w/o KNN” and “w/o SKNN” denote two WAGE variants with the K-nearest neighbor operation and the structure-oriented K-nearest neighbor operation being removed, respectively. The **boldface** value indicates the best result.

3) Influence of the DNA and HSE Arrangement: In this part, we compare two different ways of arranging the DNA and HSE modules: sequential HSE-DNA and parallel use of both modules. Specifically, “HSE-DNA-S” denotes a variant of WAGE, where the HSE module and the DNA module are arranged in a sequential manner (with DNA following HSE). “DNA-HSE-P” indicates the proposed WAGE, where the DNA

TABLE VIII
Ablation Study on the DNA and HSE Arrangement

Method	Cora			
	Recall@10	Recall@50	NDCG@10	NDCG@50
HSE-DNA-S	16.76	35.77	23.44	34.50
DNA-HSE-P	**17.28**	**36.51**	**23.84**	**35.01**
Method	Citeseer			
	Recall@10	Recall@50	NDCG@10	NDCG@50
HSE-DNA-S	8.63	24.58	14.57	26.01
DNA-HSE-P	**9.68**	**26.75**	**16.55**	**28.68**
Method	Amac			
	Recall@10	Recall@50	NDCG@10	NDCG@50
HSE-DNA-S	4.18	15.98	10.36	24.16
DNA-HSE-P	**4.47**	**16.61**	**11.08**	**24.71**
Method	Amap			
	Recall@10	Recall@50	NDCG@10	NDCG@50
HSE-DNA-S	4.27	16.30	10.55	24.28
DNA-HSE-P	**4.56**	**16.55**	**11.10**	**24.72**

Recall and NDCG using the top 10 and top 50 on four datasets are reported. “HSE DNA-S” denotes a variant of WAGE, where the HSE module and the DNA module are arranged in a sequential manner (with DNA following HSE). “DNA-HSE-P” indicates the proposed WAGE, where the DNA module and the HSE module are applied in a parallel manner. The **boldface** value indicates the best result.

module and the HSE module are applied in a parallel manner. Table VIII summarizes the experimental results of different module arranging methods. From these results, we can observe that “DNA-HSE-P” consistently achieves better performance than that of “HSE-DNA-S”. Taking the results on Citeseer for example, “DNA-HSE-P” exceeds “HSE-DNA-S” by 1.05%, 2.17%, 1.98%, and 2.67% in terms of four metrics. The observations on other datasets are similar. We attribute the superiority of “DNA-HSE-P” to the following aspect. In our weight-sharing architecture, especially for the parallel design, we consider the HSE module as an auxiliary task that enables the structure-attribute information to verify each other for establishing accurate sample correlations. Moreover, the integration of preserved latent variables from two modules promotes the incorporation of complementary information from two sources, thereby ensuring the quality of data characteristics and topology for attribute-missing samples and leading to improved missing attribute estimation. Overall, the results presented in Fig. 3 and Table VIII show that utilizing both modules is crucial, while the best-arranging strategy further facilitates performance.

E. Less Attribute-Observed Samples (RQ4)

To further investigate the superiority of WAGE, it is necessary to show whether our method can still achieve effective data completion when less visible attribute information is available. To this end, we compare the performance among SAT-GCN, SAT-GAT, and WAGE by varying the attribute-missing ratio from 20% to 80%. From the results in Fig. 4, several observations can be summarized: 1) with a relatively low attribute-missing ratio, either WAGE or SAT series of methods can achieve promising results across all datasets. Taking the WAGE method for example, for a graph neural network (GNN)-based model, it is relatively easier to learn to disregard a small portion of missing features that are initially set to zero and instead concentrate on the known features [48] for achieving effective data completion; 2) as the attribute-complete samples become fewer, it is evident that the performance of all methods would be affected to varying extents. In this circumstance, our method still demonstrates better performance than its competitors. This is because the SAT series of methods employ adversarial distribution matching to forcibly align the distributions of structure-attribute latent features in attribute-complete samples. Nonetheless, a common drawback of generative adversarial learning is the issue of discriminator over-fitting [69], [70]. This problem becomes particularly severe in scenarios with extensive missing attributes. Comparatively, we complete the missing values of the target sample by allowing the structure-attribute information to mutually enhance and validate each other within a more compact data-completing architecture; and 3) Taking the results of NDCG@10 and NDCG@50 for example, the improvement is more significant with the increase of attribute-missing ratios. The rationale behind this is that in the noisy latent space, our method not only completes missing node attributes but also mitigates the side effects of low-confidence information diffusion through a filtering-and-enhancing strategy. These results once again demonstrate that the proposed WAGE has stronger robustness against severe data absence compared to other competitors.

F. Analysis of Hyper-Parameters (RQ5)

In this subsection, we investigate the effect of four hyper-parameters, *i.e.,* number of neighbor orders O, number of nearest neighbors K, and balanced coefficients γ and λ.

1) Parameter Analysis of O and K: To illustrate the effectiveness of the DNA module in-depth, we investigate the performance variation of WAGE with respect to different setups for O and K. Specifically, in our experimental settings, we keep K fixed and systematically vary O from 1 to 9 with a step size of 2 and vice versa. From the results reported in Fig. 5, we can see that 1) taking the results on Citeseer for example, increasing the K value initially obtains promising performance but subsequently results in comparatively poorer results. These indicate that although the DNA module has proven effective in completing missing data, selecting an appropriate value for K is essential for identifying meaningful information; 2) our method can achieve superior performance when K varies from 3 to 7 across all datasets. It indicates that it is easy to find a proper parameter setting for our method across datasets. However, it is worth noting that excessively high or low K values significantly impact model performance in some cases. Taking the Cora and Amac datasets for example, on the one hand, as the number of neighbors increases to 9 in the KNN operation, the communities start to overlap, which consequently raises the risk of incorporating low-confidence features during data imputation. On the other hand, decreasing the number of neighbors to 1 diminishes the effectiveness of the KNN operation in gathering reliable features for missing attribute reconstruction; 3) increasing the O value shows a trend of first increasing and then degrading slightly. This suggests that the majority of unobserved high-confidence connections are preserved within first-order to fifth-order neighbors; and 4) in most cases, the optimal values for O and K tend to fall within the range of [3, 5]. This suggests that when one has to select a fixed O or K value for effective nearest

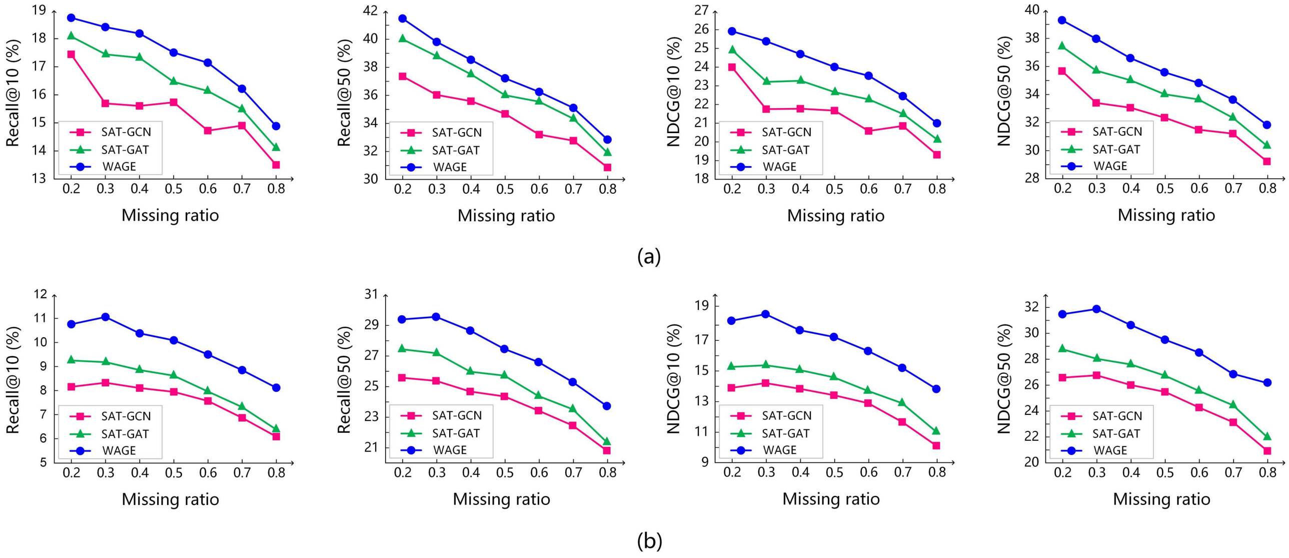


Fig. 4. Performance comparison among SAT-GCN, SAT-GAT, and the proposed WAGE with different attribute-missing ratios. Recall and NDCG using the top 10 and top 50 on Cora and Citeseer are reported.

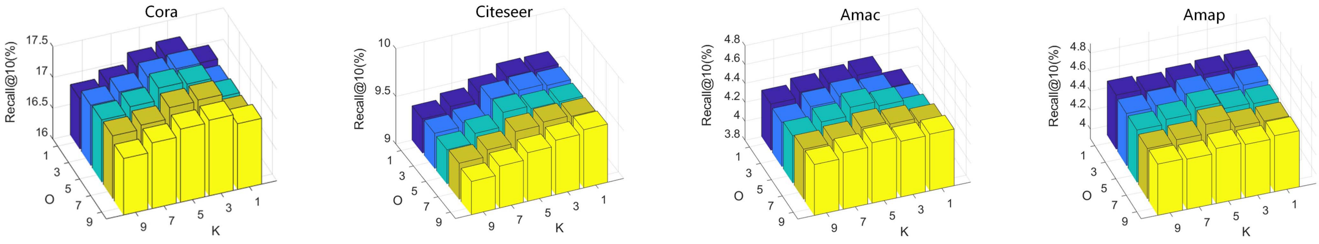


Fig. 5. Model sensitivity with the variation of hyper-parameters. The X-axis, Y-axis, and Z-axis refer to the K value, O value, and the Recall@10 performance, respectively.

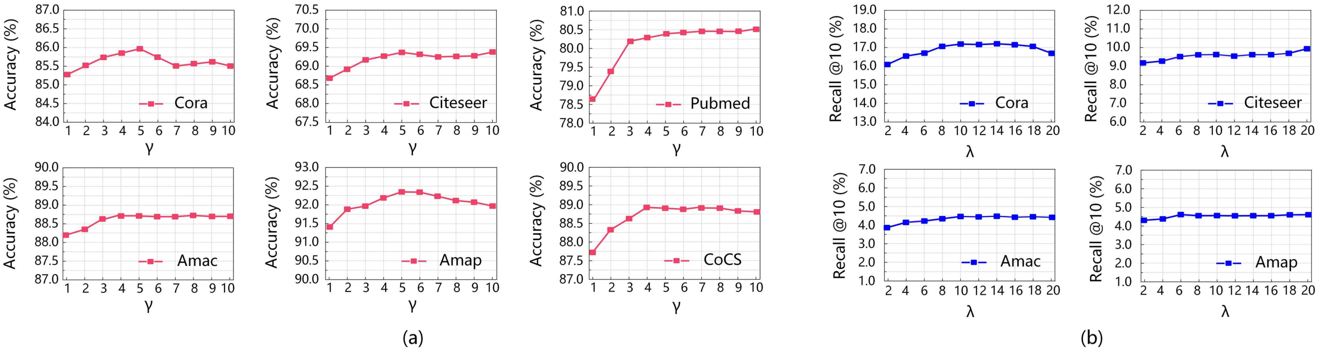


Fig. 6. Performance comparison with different hyper-parameter setups. (a) The sensitivity of WAGE when γ varies from 1 to 10 with 1 step size. The X-axis and Y-axis refer to the γ value and the accuracy performance, respectively. (b) The sensitivity of WAGE when λ varies from 2 to 20 with 2 step size. The X-axis and Y-axis refer to the λ value and the Recall@10 performance, respectively.

neighbor aggregation, choosing one within this range contributes to improved performance.

2) *Parameter Analysis of γ and λ:* As seen in (11) and (16), WAGE introduces two hyper-parameters to balance the importance of different objectives. To show their influence in-depth, we conduct an experiment to investigate the effect of γ and λ. Note that we first set one to a certain value and then tune the other carefully. Fig. 6(a) and (b) report the accuracy and Recall performance variation of WAGE when γ varies from 1 to 10 with a step size of 1 and λ varies from 2 to 20 with a step size of 2, respectively. From these sub-figures, we can observe that 1) tuning both γ and λ would cause performance variation and the model performance is more stable in the range of [3, 5] for γ and in the range of [8, 18] for λ, suggesting that searching γ and λ

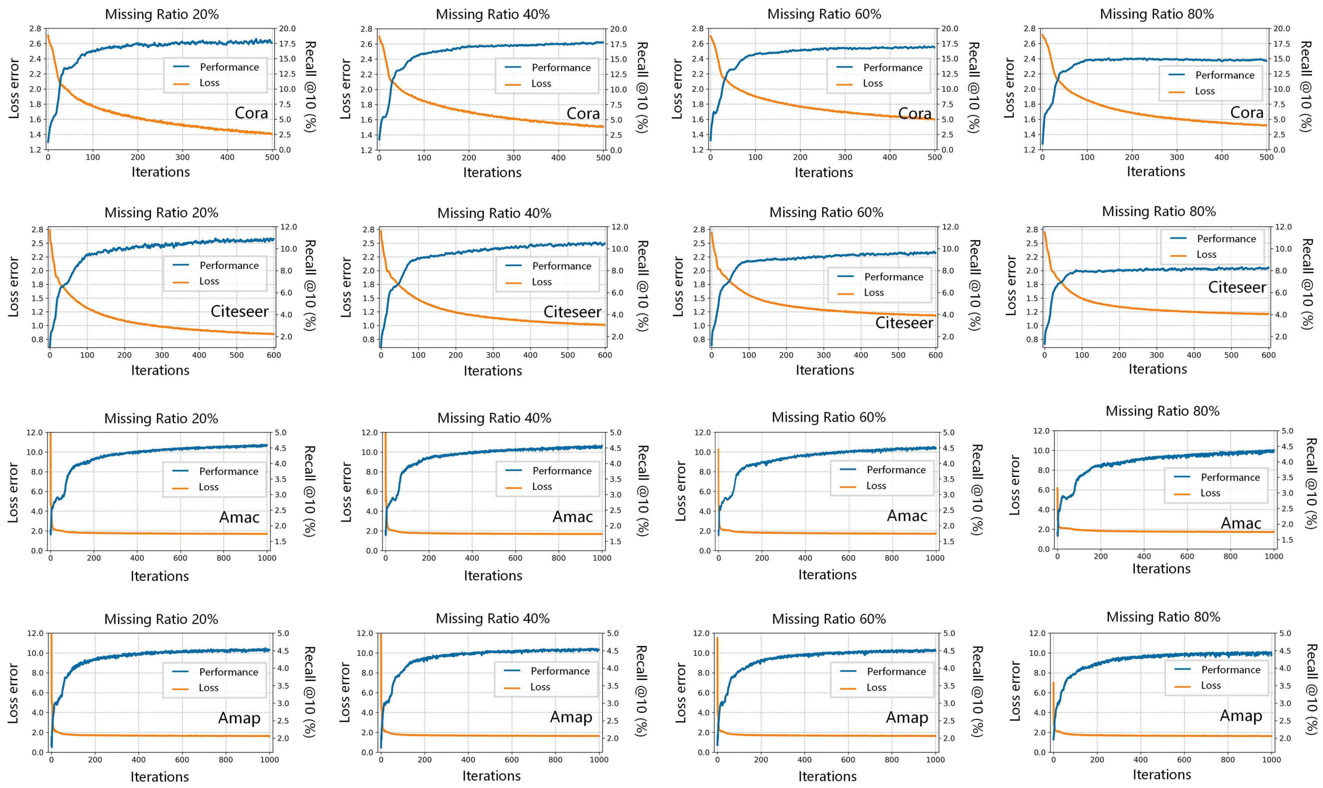


Fig. 7. Method convergence and performance variation with different attribute-missing ratios on four datasets. X-axis, left Y-axis, and right Y-axis refer to the iteration number, loss error, and Recall@10 performance, respectively.

values from a reasonable hyper-parameter region could benefit the model performance; 2) for a certain λ value, the performance exhibits a trend of first rising and then dropping slightly with the variation of γ on Cora, Amap, and CoCS. However, on Citeseer and Pubmed, the model performs better with continually increasing the value of γ. These phenomenons indicate that WAGE requires a suitable coefficient to ensure a balance between the structure-attribute reconstruction and the improvement of data completion quality. Notably, the performance of the model on Cora and Citeseer has similar trends when changing the λ value at a certain γ value; and 3) according to the results of all datasets, WAGE tends to perform well by setting γ and λ to 5 and 10, respectively.

G. *Convergence and Performance Stability (RQ6)*

This section investigates the convergence of the proposed WAGE on four datasets under various attribute-missing scenarios. As seen in Fig. 7, we record the Recall@10 performance and plot the loss error of WAGE with iterations. From these sub-figures, we can observe that 1) the Recall@10 performance of WAGE first gradually increases to a plateau with an obvious tendency and then keeps stable with a wide range of iterations; and 2) the objective monotonically decreases and WAGE usually converges in less than 1000 iterations on all datasets. These results clearly verify a good convergence property of the proposed method and reveal the stability of its learning procedure.

V. Discussion

In this section, we conduct an in-depth discussion between WAGE and some recent related works as below, including GraphMAE [53], T2-GNN [49], SAT [11], Amer [20], CSAT [18], ITR [12], FP [48], and PaGNN [54].

A. *Relation to GraphMAE*

GraphMAE [53] also reconstructs missing attribute values for graph learning. However, this method is proposed to solve representation learning for complete graph data. In its setting, attribute values are complete for each node in the input graph. This is the main difference between this method and the proposed WAGE. It is worth noting that those absent values in GraphMAE are manually made missing by randomly masking a portion of input node attributes, and the masked values are adopted as supervision to guide the network training. However, this method lacks a mechanism to handle attribute-missing graph data, its performance would decrease drastically when missing attributes exist in the input graph.

B. *Relation to T2-GNN*

T2-GNN [49] completes node attributes through knowledge distillation. T2-GNN first employs parameterized random noise to initialize missing values and conducts attribute embedding through an MLP-based teacher sub-network. It subsequently distills expert knowledge from targeted teachers into a student sub-network to provide guidance for data imputation. In the attribute-missing scenario, the teacher model struggles to extract effective representations from random noise and even aggravates the distillation of error information into the student model, significantly compromising the quality of data completion. Our method differs from T2-GNN in that it utilizes two types of K-nearest neighbor searches to aggregate the most reliable non-local features while discarding irrelevant ones, and meanwhile, designs an auxiliary task to assist in estimating missing attributes. As a result, our method can provide data imputation with more reliable guidance compared to T2-GNN when all attributes of some nodes are entirely missing.

C. Relation to SAT and Amer

For SAT [11] and Amer [20], they complete node attributes through generative adversarial learning.

- SAT first isolates the learning processes of attribute embedding and structure embedding using two decoupled sub-networks, and then conducts adversarial distribution matching to complete missing attributes by approximating the distributions of two-source latent features to that of the pre-assumed Gaussian noise.
- Amer first utilizes two independent encoders to extract the information of node attributes and graph structures, and then develops a generative adversarial learning mechanism to reconstruct missing attributes from Gaussian noise by imposing structure-attribute embedding associations.

Despite the success of recovering unknown features from scratch, generative adversarial learning has a typical drawback: the discriminator over-fitting [69], [70]. For a low attribute-missing ratio, it is not that obvious, but for a high attribute-missing ratio, this problem will be very serious. This is because the discriminator overfits the limited observed training samples, making its feedback to the generator meaningless. Besides, imposing overly strict Gaussian distribution assumptions on the latent variables would make it difficult for the model to accurately represent the true distribution of the data, inevitably leading to biased or inaccurate data completion. Unlike these methods, we complete the missing values of the target sample through a data-dependent structure-attribute mutual enhancement scheme. In the latent space, we first fill them up with the counterparts of samples that are mostly similar, and further adjust the imputed missing features by optimizing the topology relationships of reconstructed samples. By doing this, the missing values are filled up by the combination of the visible values of the most similar and credible samples in the dataset. As a consequence, the visible contents have been made the best use of.

D. Relation to CSAT and ITR

For CSAT [18] and ITR [12], they complete node attributes through heterogeneous information fusion.

- CSAT first updates the structure embedding by aggregating the attribute embedding with structural information through a graph neural network (GNN), and then conducts a hidden distribution contrast between two types of latent variables for missing attribute estimation.
- ITR first takes the structure embedding of attribute-missing samples as initial imputed information, and then refines it together with attribute embedding based on an adaptively updated affinity structure.

Although structure-attribute embeddings are intrinsically interconnected within a graph, simply integrating and diffusing heterogeneous information from two sources through affinity structure or GNN-based aggregator may lead to unpredictable results due to semantic inconsistency [71]. To solve this issue, the proposed WAGE conducts data imputation in a shared latent space. On the one hand, we closely entangle the structure-attribute information embedding processes for initial imputation through a weight-sharing encoder, taking full advantage of complementary information from the topological structure and node attributes. On the other hand, WAGE utilizes a single autoencoder design without incorporating additional parameterized encoders or decoders. With its compact network architecture, WAGE has superior storage efficiency for model size compared to competitors. As seen in Table IX, taking ITR for example, WAGE reduces over 47.8%, 73.0%, 91.7%, 86.4%, 97.8%, and 83.3% model parameters compared to ITR on Cora, Citeseer, Amac, Amap, Pubmed, and CoCS, respectively.

TABLE IX
NETWORK PARAMETER (MILLION) COMPARISON ON SIX BENCHMARK DATASETS

Method	Cora	Citeseer	Amac	Amap	Pubmed	CoCS
SAT-GCN	1.17	2.20	3.11	1.88	4.20	6.45
SAT-GraphSage	1.72	2.88	5.87	3.42	8.52	10.13
SAT-GAT†	1.17	2.20	3.11	1.88	4.20	6.45
ITR	1.15	5.59	7.92	4.78	10.71	16.45
WAGE (ours)	0.60	1.51	0.66	0.65	0.23	2.75

† SAT-GAT adopts one-head attention to reduce training cost, as was done in the original paper [11].

Note that the parameter numbers of Amer and CSAT are unavailable due to inaccessible source code.

E. Relation to FP and PaGNN

FP [48] and PaGNN [54] completes node attributes through feature propagation. Specifically, FP and PaGNN reconstruct the missing node features by iteratively diffusing the visible features within the graph. For this type of method, some unreliable features would diffuse through the feature propagation process and then overwhelm the reconstructed graph. However, FP and PaGNN lack a mechanism to alleviate low-confidence information diffusion, reducing the reliability of data imputation. In contrast, WAGE introduces a new noise filtering and information enhancing strategy to optimize the initially imputed data. In our strategy, we introduce a K-nearest neighbor-based dual non-local learning mechanism to improve the quality of data imputation by revealing unobserved high-confidence connections while filtering unreliable ones. Besides, we have structure-attribute information to verify each other and enable the network to focus more on enhancing the topology relationships of reconstructed attribute-missing samples through a weighted edge reconstruction scheme. Consequently, more comprehensive data completion is obtained.

In summary, the differences between WAGE and recent related works lie in the following main aspects: data completion, network architecture design, data distribution assumption, and model performance. Our newly designed missing attribute imputation mechanism brings about the following advantages: more comprehensive data completion, more compact data-completing architecture, free of data distribution assumption, and better performance.

VI. Conclusion and Future Work

In this paper, we propose a novel weight-sharing framework termed WAGE to process attribute-missing graphs, which enjoys the merits of more comprehensive data completion, more compact data-completing architecture, and free of prior assumptions. The proposed dual non-local aggregating and hidden structure enhancing modules can lead to high-quality attribute imputation and accurate topology reconstruction towards improved data completion. Extensive experiments on six benchmark datasets have demonstrated the effectiveness and superiority of WAGE against state-of-the-art competitors. We expect that the merits of WAGE will make it a go-to solution for practical attribute-missing graph analysis applications. Furthermore, we outline a few potential failure cases that could arise: 1) hybrid-absent graph data. Our graph imputation method heavily relies on trustworthy and complete structure information to conduct graph representation learning with graph neural networks (GNNs). However, it is widely acknowledged that GNNs face challenges in generalizing well when dealing with both incomplete topology and missing nodes. Consequently, this limitation may lead to inaccurate imputation and unrepresentative graph embeddings, potentially affecting the performance of the proposed WAGE; and 2) high data sparsity. Our method may encounter difficulties in completing missing graph data in scenarios where the connections between nodes are highly sparse. The limited interactions among entities can impede the model's ability to accurately infer missing content based on available observations, potentially affecting the quality of data completion. Future work may extend WAGE to handle hybrid-absent graphs, explore further applications, and improve initial imputation using automated techniques [72], [73]. Additionally, exploring if WAGE can work well in other cases, such as highly sparse graphs, is another interesting direction.

References

[1] X. Shu, B. Xu, L. Zhang, and J. Tang, "Multi-granularity anchor-contrastive representation learning for semi-supervised skeleton-based action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7559–7576, Jun. 2023.

[2] C. Chen, Y. Liu, L. Chen, and C. Zhang, "Bidirectional spatial-temporal adaptive transformer for urban traffic flow forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 6913–6925, Oct. 2023.

[3] Y. Xie, Z. Xu, J. Zhang, Z. Wang, and S. Ji, "Self-supervised learning of graph neural networks: A unified review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2412–2429, Feb. 2023.

[4] R. Zhang, Y. Zhang, C. Lu, and X. Li, "Unsupervised graph embedding via adaptive graph learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5329–5336, Apr. 2023.

[5] J. Melton and S. Krishnan, "muxGNN: Multiplex graph neural network for heterogeneous graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 11067–11078, Sep. 2023.

[6] X. Zhang, D. Song, and D. Tao, "Hierarchical prototype networks for continual graph representation learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4622–4636, Apr. 2023.

[7] K. Liang et al., "A survey of knowledge graph reasoning on graph types: Static, dynamic, and multi-modal," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 9456–9478, Dec. 2024.

[8] K. Liang et al., "Knowledge graph contrastive learning based on relation-symmetrical structure," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 1, pp. 226–238, Jan. 2024.

[9] I. Spinelli, S. Scardapane, and A. Uncini, "Missing data imputation with adversarially-trained graph convolutional networks," *Neural Netw.*, vol. 129, pp. 249–260, 2020.

[10] H. Taguchi, X. Liu, and T. Murata, "Graph convolutional networks for graphs containing missing features," *Future Gener. Comput. Syst.*, vol. 117, pp. 155–168, 2021.

[11] X. Chen, S. Chen, J. Yao, H. Zheng, Y. Zhang, and I. W. Tsang, "Learning on attribute-missing graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 740–757, Feb. 2022.

[12] W. Tu et al., "Initializing then refining: A simple graph attribute imputation network," in *Proc. Int. Joint Conf. Artif. Intell.*, 2022, pp. 3494–3500.

[13] F. Monti, M. M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3697–3707.

[14] L. Zheng, C.-T. Lu, F. Jiang, J. Zhang, and P. S. Yu, "Spectral collaborative filtering," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 311–319.

[15] C. Gao et al., "Neural multi-task recommendation from multi-behavior data," in *Proc. IEEE Int. Conf. Data Eng.*, 2019, pp. 1554–1557.

[16] C. Li, S. Wang, D. Yang, P. S. Yu, Y. Liang, and Z. Li, "Adversarial learning for multi-view network embedding on incomplete graphs," *Knowl.-Based Syst.*, vol. 180, pp. 91–103, 2019.

[17] J. You, X. Ma, D. Y. Ding, M. J. Kochenderfer, and J. Leskovec, "Handling missing data with graph representation learning," in *Proc. Adv. Conf. Neural Inf. Process. Syst.*, 2020, pp. 19075–19087.

[18] M. Li, Y. Zhang, W. Zhang, S. Zhao, X. Piao, and B. Yin, "CSAT: Contrastive sampling-aggregating transformer for community detection in attribute-missing networks," *IEEE Trans. Comput. Social Syst.*, vol. 11, no. 2, pp. 2277–2290, Apr. 2024.

[19] J. Yoo, H. Jeon, J. Jung, and U. Kang, "Accurate node feature estimation with structured variational graph autoencoder," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 2336–2346.

[20] D. Jin et al., "Amer: A new attribute-missing network embedding approach," *IEEE Trans. Cybern.*, vol. 53, no. 7, pp. 4306–4319, Jul. 2023.

[21] Y. Shi, J. Xi, D. Hu, Z. Cai, and K. Xu, "RayMVSNet++ : Learning ray-based 1D implicit fields for accurate multi-view stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13666–13682, Nov. 2023.

[22] T. Zhou, Z. Cai, F. Liu, and J. Su, "In pursuit of beauty: Aesthetic-aware and context-adaptive photo selection in crowdsensing," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 9364–9377, Sep. 2023.

[23] S. Zhao, M. Hu, Z. Cai, H. Chen, and F. Liu, "Dynamic modeling cross- and self-lattice attention network for chinese NER," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 14515–14523.

[24] S. Zhao, M. Hu, Z. Cai, and F. Liu, "Modeling dense cross-modal interactions for joint entity-relation extraction," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, 2020, pp. 4032–4038.

[25] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.

[26] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.

[27] Z. Wang, Z. Li, R. Wang, F. Nie, and X. Li, "Large graph clustering with simultaneous spectral embedding and discretization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4426–4440, Dec. 2021.

[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[29] W. Tu et al., "Deep fusion clustering network," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 9978–9987.

[30] Y. Liu et al., "Deep graph clustering via dual correlation reduction," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7603–7611.

[31] Y. Xie, Z. Xu, and S. Ji, "Self-supervised representation learning via latent graph prediction," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 24460–24477.

[32] Y. Yang et al., "Self-supervised heterogeneous graph pre-training based on structural clustering," in *Proc. Conf. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 16962–16974.

[33] L. Gong, W. Tu, S. Zhou, L. Zhao, Z. Liu, and X. Liu, "Deep fusion clustering network with reliable structure preservation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 7792–7803, Jun. 2024.

[34] Y. Liu et al., "Hard sample aware network for contrastive deep graph clustering," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 8914–8922.

[35] D. Hu, K. Liang, S. Zhou, W. Tu, M. Liu, and X. Liu, "scDFC: A deep fusion clustering method for single-cell RNA-seq data," *Brief. Bioinf.*, vol. 24, no. 4, pp. 1–9, 2023.

[36] W. Tu et al., "RARE: Robust masked graph autoencoder," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 10, pp. 5340–5353, Oct. 2024.

[37] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proc. Int. Conf. Learn. Representations*, 2019.

[38] Z. Peng et al., "Learning representations by graphical mutual information estimation and maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 722–737, Jan. 2023.

[39] K. Hassani and A. H. K. Ahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4116–4126.

[40] L. Gong, S. Zhou, W. Tu, and X. Liu, "Attributed graph clustering with dual redundancy reduction," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2022, pp. 3015–3021.

[41] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *Proc. Int. World Wide Web Conf.*, 2021, pp. 2069–2080.

[42] W. Tu, S. Zhou, X. Liu, C. Ge, Z. Cai, and Y. Liu, "Hierarchically contrastive hard sample mining for graph self-supervised pre-training," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 16748–16761, Nov. 2024.

[43] X. Xu, C. Deng, Y. Xie, and S. Ji, "Group contrastive self-supervised learning on graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3169–3180, Mar. 2023.

[44] S. Qian, D. Xue, Q. Fang, and C. Xu, "Integrating multi-label contrastive learning with dual adversarial graph neural networks for cross-modal retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4794–4811, Apr. 2023.

[45] W. Tu et al., "Attribute-missing graph clustering network," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 15392–15401.

[46] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*.

[47] M. Zhang and Y. Chen, "Inductive matrix completion based on graph neural networks," in *Proc. Int. Conf. Learn. Representations*, 2020.

[48] E. Rossi, H. Kenlay, M. I. Gorinova, B. P. Chamberlain, X. Dong, and M. M. Bronstein, "On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features," in *Proc. Learn. Graphs Conf.*, 2022, pp. 1–11.

[49] C. Huo, D. Jin, Y. Li, D. He, Y.-B. Yang, and L. Wu, "T2-GNN: Graph neural networks for graphs with incomplete features and structure via teacher-student distillation," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 4339–4346.

[50] D. Jin, C. Huo, C. Liang, and L. Yang, "Heterogeneous graph neural network via attribute completion," in *Proc. Int. World Wide Web Conf.*, 2021, pp. 391–400.

[51] D. He et al., "Analyzing heterogeneous networks with missing attributes by unsupervised contrastive learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4438–4450, Apr. 2024.

[52] H. Cui, Z. Lu, P. Li, and C. Yang, "On positional and structural node features for graph neural networks on non-attributed graphs," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 3898–3902.

[53] Z. Hou et al., "GraphMAE: Self-supervised masked graph autoencoders," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 594–604.

[54] B. Jiang and Z. Zhang, "Incomplete graph representation and learning via partial graph neural networks," 2020, *arXiv:2003.10130*.

[55] M. Liu, Z. Wang, and S. Ji, "Non-local graph neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 10270–10276, Dec. 2022.

[56] Y. Liu, S. Yang, Y. Xu, C. Miao, M. Wu, and J. Zhang, "Contextualized graph attention network for recommendation with item knowledge graph," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 181–195, Jan. 2023.

[57] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, *arXiv:1611.07308*.

[58] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Inf. Retrieval*, vol. 3, no. 2, pp. 127–163, 2000.

[59] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *Inf. Retrieval*, vol. 29, no. 3, pp. 93–106, 2008.

[60] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, "Pitfalls of graph neural network evaluation," in 2018, *arXiv:1811.05868*.

[61] G. Namata, B. London, L. Getoor, and B. Huang, "Query-driven active surveying for collective classification," in *Proc. Conf. Neural Inf. Process. Syst.*, 2012, p. 1.

[62] Özgür Simsek and D. D. Jensen, "Navigating networks by using homophily and degree," *Nat. Acad. Sci.*, vol. 105, no. 35, pp. 12758–12762, 2008.

[63] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[64] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.

[65] L. Hu, S. Jian, L. Cao, Z. Gu, Q. Chen, and A. Amirbekyan, "Hers: Modeling influential contexts with heterogeneous relations for sparse and cold-start recommendation," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3830–3837.

[66] X. Huang, Q. Song, Y. Li, and X. Hu, "Graph recurrent networks with attributed random walks," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2019, pp. 732–740.

[67] L. Chen, S. Gong, J. Bruna, and M. M. Bronstein, "Attributed random walk as matrix factorization," in *Proc. Conf. Neural Inf. Process. Syst. Workshop*, 2019.

[68] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. Int. Conf. Learn. Representations*, 2014.

[69] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 12104–12114.

[70] K. Cui, J. Huang, Z. Luo, G. Zhang, F. Zhan, and S. Lu, "GENCO: Generative co-training for generative adversarial networks with limited data," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 499–507.

[71] Y. Sun, D. Zhu, H. Du, and Z. Tian, "MHNF: Multi-hop heterogeneous neighborhood information fusion graph representation learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 7192–7205, Jul. 2023.

[72] W. Jin, X. Liu, X. Zhao, Y. Ma, N. Shah, and J. Tangi, "Automated self-supervised learning for graphs," in *Proc. Int. Conf. Learn. Representations*, 2022.

[73] Y. Wang, J. Lin, J. Zou, Y. Pan, T. Yao, and T. Mei, "Improving self-supervised learning with automated unsupervised outlier arbitration," in *Proc. Conf. Neural Inf. Process. Syst.*, 2021, pp. 27617–27630.

Wenxuan Tu received the PhD degree in the School of Computer, National University of Defense, Changsha, China, in 2023. He is currently an associate professor with the School of Computer Science and Technology, Hainan University, Haikou, China. His current research interests include clustering analysis, multi-view learning, and graph machine learning. He has authored or coauthored more than 30 papers in highly regarded journals and conferences, such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-SMC, NeurIPS, ICML, ICLR, AAAI, IJCAI, CVPR, ACM MM.

Sihang Zhou (Member, IEEE) received the BS degree in information and computing science, the MS degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2014 and 2012, respectively, and the PhD degree from the National University of Defense Technology (NUDT), Changsha, in 2019. He is currently an Associate professor with the School of Intelligence Science and Technology, NUDT. His current research interests include machine learning, knowledge graphs, and medical image analysis.

Xinwang Liu (Senior Member, IEEE) received the PhD degree from the National University of Defense Technology (NUDT), Changsha, China, in 2013. He is currently a full professor with the School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. He has authored or coauthored more than 200 peer-reviewed papers, including those in highly regarded journals and conferences, such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-MM, IEEE T-IFS, ICML, NeurIPS, ICCV, CVPR, AAAI, and IJCAI. He is an associate editor for *the Information Fusion Journal*, *IEEE Transactions on Cybernetics*, and *IEEE Transactions on Neural Networks and Learning Systems* Journal.

Zhiping Cai received the BS, MS, and PhD degrees in computer science and technology from the National University of Defense Technology (NUDT), Changsha, China, in 1996, 2002, and 2005, respectively. He is currently a full professor with the School of Computer, NUDT. His current research interests include artificial intelligence, network security, and Big Data. Prof. Cai is currently a senior member of CCF.

Yawei Zhao received the PhD degree in computer science from the National University of Defense Technology, China, in 2020. He is currently with Medical Big Data Research Center, Chinese PLA General Hospital, Beijing, China. His research interests include time-series analysis, medical data analysis, and federated learning.

Yue Liu (Member, IEEE) received the BS degree from Northeastern University, Qinhuangdao, China, in 2021. and the MS degree from the National University of Defense Technology, Changsha, China,in 2023. He is currently working toward the PhD degree with the National University of Singapore. He has authored or coauthored more than 30 peer-reviewed papers, including ICML, NeurIPS, ICLR, and T-PAMI. His current research interests include graph neural networks, deep clustering, and recommendation systems.

Kunlun He received the MD degree from The 3rd Military Medical University, Chongqing, China, in 1988, and the PhD degree in cardiology from Chinese PLA Medical School, Beijing, China in 1999. He was a postdoctoral research fellow with the Division of Circulatory Physiology of Columbia University, from 1999 to 2003. He is currently the Director and Professor of the Medical Big Data Research Center, Chinese PLA General Hospital. His research interests include Big Data and artificial intelligence of cardiovascular disease.