

# Revisiting Initializing Then Refining: An Incomplete and Missing Graph Imputation Network

Wenxuan Tu, Bin Xiao, Xinwang Liu, *Senior Member, IEEE*, Sihang Zhou, Zhiping Cai, and Jieren Cheng

**Abstract**—With the development of various applications, such as social networks and knowledge graphs, graph data has been ubiquitous in the real world. Unfortunately, graphs usually suffer from being absent due to privacy-protecting policies or copyright restrictions during data collection. The absence of graph data can be roughly categorized into attribute-incomplete and attribute-missing circumstances. Specifically, attribute-incomplete indicates that a part of the attribute vectors of all nodes are incomplete, while attribute-missing indicates that the whole attribute vectors of partial nodes are missing. Although many graph imputation methods have been proposed, none of them is custom-designed for a common situation where both types of graph data absence exist simultaneously. To fill this gap, we develop a novel graph imputation network termed Revisiting Initializing Then Refining (RITR), where we complete both attribute-incomplete and attribute-missing samples under the guidance of a novel initializing-then-refining imputation criterion. Specifically, to complete attribute-incomplete samples, we first initialize the incomplete attributes using Gaussian noise before network learning, and then introduce a structure-attribute consistency constraint to refine incomplete values by approximating a structure-attribute correlation matrix to a high-order structural matrix. To complete attribute-missing samples, we first adopt structure embeddings of attribute-missing samples as the embedding initialization, and then refine these initial values by adaptively aggregating the reliable information of attribute-incomplete samples according to a dynamic affinity structure. To the best of our knowledge, this newly designed method is the first end-to-end unsupervised framework dedicated to handling hybrid-absent graphs. Extensive experiments on four datasets have verified that our methods consistently outperform existing state-of-the-art competitors.

**Index Terms**—incomplete multi-view learning, graph neural network, hybrid-absent data, feature completion.

## I. INTRODUCTION

**G**RAPHS, which model and represent the complicated relationships among real-world objects, are ubiquitous in practical scenarios, including citation graphs, social graphs, protein graphs, and molecule graphs. To analyze the graph data, graph machine learning attempts to transform an original graph into low-dimensional sample representations by preserving node attributes and graph structure simultaneously [1]–[3]. In recent years, with the help of graph neural networks

W. Tu, X. Liu, and Z. Cai, are with the College of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: {wenxuantu}@163.com, {xinwangliu,zpcai}@nudt.edu.cn).

B. Xiao is with the Department of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: xiaobin@cqupt.edu.cn).

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

J. Cheng is with the School of Computer Science and Technology, Hainan University, Haikou, 570228 China, with the Hainan blockchain technology engineering research center, Haikou 570228, China (E-mail: cjr22@163.com).

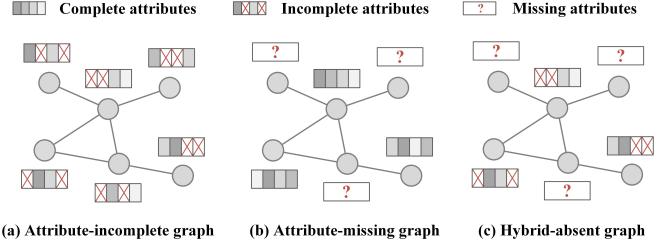


Fig. 1. Different types of absent graphs. (a) Attribute-incomplete graph: particular attributes of all samples are absent; (b) Attribute-missing graph: all attributes of specific nodes are absent; (c) Hybrid-absent graph: both circumstances (a) and (b) exist simultaneously within a graph. As one can easily see, the last category is the most challenging, however, it is still under-explored in previous literature. We make the first attempt to solve it by proposing a novel method called RITR.

(GNNs), graph machine learning has become an increasingly powerful artificial intelligence technique. It has achieved significant success in diverse real-world applications, such as anomalous citation detection [4], few-shot learning [5], feature selection [6], and knowledge graphs [7].

The key prerequisite for the impressive performance of existing graph machine learning methods lies in the assumption that all samples within a graph are available and complete. However, this assumption may not always hold in practice since it is hard to collect all information from graph data. The reason behind including but not limited to privacy-protecting policies, copyright restrictions, and simply not enough information. For example, in a co-purchase graph of Amazon, consumers tend to selectively (or entirely not) provide their feedback for specific items due to privacy concerns. In a citation network, some papers are inaccessible due to copyright protection. All these circumstances could easily trigger sparsity and data-absent problems that adversely affect graph representations. According to the type of node attribute absence, the absent graphs can be roughly divided into two categories: 1) the attribute-incomplete graph where only a portion of attributes of all nodes are absent; 2) the attribute-missing graph where all attributes of specific nodes are absent. Fig. 1 illustrates the situations of attribute absence. Among them, Fig. 1(a) corresponds to an attribute-incomplete graph, Fig. 1(b) corresponds to an attribute-missing graph, and Fig. 1(c) corresponds to a hybrid-absent graph where both attribute-incomplete samples and attribute-missing samples exist in the same graph. The above cases make valuable information invisible and pose significant challenges to existing graph machine learning methods for graph analysis.

To solve the attribute-incomplete learning problem, many

efforts have been devoted to developing various imputation strategies such as matrix completion [8], [9], generative adversarial network [10], Gaussian mixture model (GMM) [11], and other advanced ones [12], [13]. With the imputed attributes, these methods then integrate a standard GNN-based framework with data imputation techniques to conduct the sample embedding. Although significant progress has been made in solving the attribute-incomplete learning problem, the performance of these methods degrades drastically when they handle extremely absent data (*e.g.*, attribute-missing graphs). To solve this problem, a recent advanced method termed SAT [14] first introduces an unsupervised graph imputation framework to handle attribute-missing graphs under the guidance of a shared-latent space assumption. Specifically, SAT utilizes a graph neural network (*e.g.*, GCN [15] or GAT [16]) to embed the available attributes and graph structure into a latent space in a decoupled manner. Then it performs a distribution matching mechanism to recover the unknown values of attribute-missing samples. Although achieving encouraging success, SAT also suffers from the following limitations when conducting the data imputation: 1) two-source information isolation. SAT isolates the learning processes of embeddings of observed attributes and the complete graph structure. This prevents the trustworthy visible information from being sufficiently utilized, which could cause the learned representations to be biased and also increase the risk of inaccurate data imputation; 2) strict prior assumption. SAT forces two-source latent variables to align with an in-discriminative noise matrix obeying normal distribution. While in reality, the pre-defined normal distribution would not ideally conform to the complex graphs. As a result, the negotiation between attribute and structure information tends to get overly rigid, resulting in less discriminative representations. This could adversely affect the quality of the rebuilt attribute matrix of all samples, especially those without attributes.

To overcome the above issues, we propose an Initializing Then Refining (ITR) network [17] to forbid the adverse effect of inaccurate simple initialization and the limitation of rigid distribution assumption. The core idea of ITR is to fully leverage the trustworthy visible information to implement the sample embeddings for missing attribute imputation. Though being the potential for better tackling the attribute-missing problem, we observe that when ITR processes hybrid-absent graphs, attribute-incomplete samples would largely undermine the quality of generated attribute-missing features due to the diffusion of inaccurate imputed information. To our knowledge, hybrid-absent graph machine learning has not been studied in the existing graph literature, which is a universal and more challenging problem for various practical applications. To fill this gap, we revisit ITR and further improve it by designing a variant, termed Revisiting Initializing Then Refining (RITR). In this newly proposed RITR, we complete both attribute-incomplete and attribute-missing samples under the guidance of the initializing-then-refining imputation criterion.

To impute incomplete attributes, we elaborately design a Sample-denoising Then Consistency-preserving (STC) mechanism. As illustrated in Fig. 2, the feature completion process within this component mainly includes step 1 to step 3. Firstly,

we learn the sample embeddings through a denoising learning approach by combining the attribute and structure information of attribute-incomplete samples sufficiently. Secondly, we take all nodes and learn sample embeddings only according to the structure information. Finally, we explicitly develop a structure-attribute consistency constraint to refine these incomplete latent variables by approximating a structure-attribute correlation matrix to a high-order structure matrix. This operation aims to guarantee the representation quality of nodes with incomplete attributes, and meanwhile provide a feasible initialization to those nodes with missing attributes in the next step. To impute missing attributes, we design another data imputation mechanism termed Initializing Then Refining (ITR). In this component, we first take the structure embeddings of attribute-missing samples as initial imputed variables and then refine them with an adaptively updated affinity structure for embedding refinement. The above operations correspond to step 4 and step 5 in Fig. 2. Comprehensive experiments on four benchmark datasets have been conducted to verify the effectiveness and superiority of our proposed methods and components. As demonstrated, ITR consistently outperforms state-of-the-art methods. Moreover, the other proposed variant, *i.e.*, RITR, further improves the profiling and classification performance against ITR. It is expected that the simplicity and effectiveness of the RITR method will make it a promising option to be considered for practical applications where the hybrid-absent case is encountered.

This work is a substantially extended version of our original conference paper [17]. Compared with its previous version, it has the following significant improvements. 1) ***Novel research problem.*** To the best of our knowledge, hybrid-absent graph machine learning is a rarely explored research field yet is a real-world demand from various applications. Accordingly, we develop a novel graph machine learning framework called RITR without relying on any pre-assumed distribution assumption, which is the first incomplete and missing graph imputation network to solve the corresponding learning problem. 2) ***Newly proposed strategy.*** To complete attribute-incomplete samples, we propose a new feature completion mechanism termed STC by following the initializing-then-refining imputation criterion. This operation enables the model to generate more robust and discriminative features for attribute-incomplete samples, best serving subsequent attribute-missing imputation tasks. 3) ***More experimental results and analyses.*** Besides more detailed discussion and extension, we also conduct more comprehensive experiments, and all evaluation results have verified that the two proposed methods achieve the best performance in different absent graph situations.

The remainder of this paper is organized as follows. Section 2 reviews related work in terms of unsupervised graph machine learning and graph machine learning on absent graphs. Section 3 presents the notations, definitions, network and component design, and learning targets. Section 4 conducts experiments and discusses the results. Section 5 draws a final conclusion.

## II. RELATED WORK

### A. Unsupervised Graph Machine Learning

Early solutions to unsupervised graph machine learning mainly focus on random walk-based methods [18]–[20], which first generate the random walk sequences over the network structure properties and then utilize a Skip-Gram model to learn graph representations. However, these methods heavily rely on structure information and overlook other available properties (*e.g.*, attribute information) in the graph. More recently, since the powerful neighborhood aggregation capacity of graph neural networks (GNNs), many efforts have been made to design GNN-based methods. As one of the most representatives, generative/predictive learning-oriented methods aim to explore abundant information embedded in data via some well-known approaches, such as auto-encoder learning [21]–[24] and adversarial learning [25]–[28]. Another line pays attention to graph contrastive learning, which aims to maximize the agreement of two jointly sampled positive pairs [29]–[34]. One underlying assumption commonly adopted by these methods is that the attributes of all nodes are trustworthy and complete. While in real-world scenarios, they may suffer from significant performance degradation when handling absent graphs.

### B. Graph Machine Learning on Absent Graphs

According to the type of absent graphs, we can roughly group existing absent graph machine learning methods into the following three categories.

1) *Attribute-incomplete Graph Machine Learning*: In the attribute-incomplete circumstance, some methods propose to leverage data imputation-oriented techniques to restore the incomplete information, such as matrix completion [8], generative adversarial network (GAN) [35], and Gaussian mixture model (GMM) [36]. For instance, NMTR [37] and GRAPE [38], two typical matrix completion methods, first take the user-item rating matrix, users (or items), and the observed ratings as a bipartite graph, sample attributes, and connected relationships, respectively. These methods then adopt a graph neural network to predict the probabilities (regarded as imputed values) of absent connected relationships. Similarly, GRAPE [38] first converts a data imputation task into a linkage prediction learning process over a created bipartite graph, and then utilizes a graph neural network to solve it. Recent efforts like NMTR [37] and IGMC [9] follow the same paradigm as previous matrix completion methods to conduct the data imputation and sample embedding in a transductive or inductive learning manner. In addition, GINN [10] first initializes the incomplete values by a binary mask matrix before network training, and then learns a graph neural network with an adversarial learning mechanism to complete the absent information. GCNMF [11] utilizes a Gaussian mixture model to estimate the incomplete features according to the available information, and in the meanwhile, jointly optimizes the Gaussian mixture model and graph neural network in a united framework. More recently, T2-GNN [39] designs a general teacher-student graph learning framework to restore

both incomplete node features and graph structure through distillation.

2) *Attribute-missing Graph Machine Learning*: Compared to the attribute-incomplete circumstance, handling the graph data with a majority of samples having no attributes poses more challenges in learning high-quality node representations. This topic has attracted great attention from graph machine learning researchers recently. For example, HGNN-AC [40] first adopts current heterogeneous information networks (HINs) to learn node topological representations, and then utilizes the topological relationship between nodes as guidance to implement feature completion for attribute-missing samples via an attention mechanism. HGCA [41] is an unsupervised heterogeneous graph contrastive learning approach for heterogeneous graphs with missing attributes. It employs the contrastive learning technique to unify the processes of feature completion and representation learning, and thereafter, conduct a fine-grained attribute completion by extracting the semantic relations among different types of samples. Besides the attribute-missing heterogeneous graph machine learning, an advanced method called SAT [14] makes the first attempt to solve the attribute-missing learning problem over the homogeneous graphs. By unifying the data imputation and network learning processes into a single optimization procedure, SAT learns two-source information embedding matrices in a decoupled manner and then aligns them with a noise matrix sampled from a normal distribution for attribute restoration. Another recent work, ITR [17] introduces an initializing-then-refining mechanism, enabling the network to fully use the trustworthy visible information to adaptively conduct the sample embedding for missing attribute imputation. More recently, SVGA [42] and Amer [43] develop an auto-encoder-style framework to estimate missing node features via structured variational inference and adversarial learning techniques, respectively.

3) *Hybrid-absent Graph Machine Learning*: As aforementioned, attribute-incomplete and attribute-missing graph machine learning problems have been intensively studied in recent years. Despite their significant progress, in nature, these methods are not capable of effectively handling hybrid-absent graphs. In this circumstance, especially for the unsupervised scenario, the performance of existing attribute-incomplete and attribute-missing methods could drop drastically since they suffer from at least one of the following limitations: 1) heavily relying on annotated graph data; 2) lacking a specialized feature completion mechanism for handling attribute-missing (or attribute-incomplete) samples; 3) disconnecting the processes of data imputation and network optimization; 4) isolating the learning processes of structure and attribute embeddings; 5) imposing too strict a distribution assumption on the latent variables. Although our recently proposed ITR could address most of the above problems and exhibits powerful learning capacity in the attribute-missing situation, it is still a great challenge to recover incomplete and missing values with limited available information simultaneously. To achieve this goal, we study a new important research problem termed hybrid-absent graph machine learning, and further improve ITR by designing another variant called RITR. It proposes to first leverage the intimate structure-attribute relationship to

guide the imputation of incomplete attributes and then employ the most trustworthy visible information to implement the missing attribute completion. To the best of our knowledge, none of the above literature considers the hybrid-absent graph machine learning problem. RITR is the first work dedicated to this field.

### III. APPROACH

Fig. 2 shows an overview of our proposed RITR. As follows, we will provide details on notations, definitions, crucial components, and learning targets, respectively.

#### A. Notations and Definitions

$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  denotes a given undirected graph that contains  $N$  samples with  $C$  categories, where  $\mathcal{V}$  and  $\mathcal{E}$  indicate the node set and edge set, respectively. Generally, the topology of a graph  $\mathcal{G}$  can be characterized by its adjacency matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$  and the content of graph  $\mathcal{G}$  can be represented by its attribute matrix  $\mathbf{X} \in \mathbb{R}^{N \times D}$ , where  $D$  refers to the sample dimension. The main notations and their explanations are summarized in Table I.

**Definition 1 (Hybrid-absent Graph).** We denote a hybrid-absent graph  $\tilde{\mathcal{G}} = \{\mathcal{V}^I, \mathcal{V}^M, \mathcal{E}\}$ , where partial attributes of some samples are unavailable (*i.e.*, the attribute-incomplete sample set  $\mathcal{V}^I$ ) and all attributes of other samples are entirely missing (*i.e.*, the attribute-missing sample set  $\mathcal{V}^M$ ).  $N^I = |\mathcal{V}^I|$  and  $N^M = |\mathcal{V}^M|$  refer to the number of attribute-incomplete samples and attribute-missing samples, respectively. Accordingly,  $\mathcal{V} = \mathcal{V}^I \cup \mathcal{V}^M$ ,  $\mathcal{V}^I \cap \mathcal{V}^M = \emptyset$ , and  $N = N^I + N^M$ . Note that the structure information (*i.e.*,  $\mathcal{E}$ ) of  $\tilde{\mathcal{G}}$  is complete.

**Definition 2 (Learning Task).** In this work, we mainly focus on addressing the hybrid-absent graph machine learning problem on graphs without label annotation. Our auto-encoder-style framework works for learning two graph encoding functions  $E_A(\cdot)$  and  $E_s(\cdot)$  to impute invisible latent variables. Then a graph encoding function  $D(\cdot)$  will recover the attribute-missing and attribute-incomplete samples based on the imputed hidden features. The recovered attributes can be saved and used for profiling and node classification tasks.

#### B. Overview

To tackle the issue that existing studies fail to perform well on hybrid-absent graphs, we introduce an end-to-end unsupervised graph imputation network termed RITR. Our goal is to design personalized incomplete and missing feature completion mechanisms on hybrid-absent graphs, and achieve accurate data imputation and effective information propagation with the initializing-then-refining imputation criterion. As illustrated in Fig. 2, RITR contains two core components, *i.e.*, Sample-denoising Then Consistency-preserving (STC) mechanism (Section III-C) and Initializing Then Refining (ITR) mechanism (Section III-D), which are intended to solve the attribute-incomplete and attribute-missing problems, respectively. Specifically, before network learning, we first generate a corrupted sub-graph by randomly adding Gaussian noise information to incomplete attributes as initial values. Then the

TABLE I  
SUMMARY OF MAIN NOTATIONS

| Notation                                                  | Explanation                                                   |
|-----------------------------------------------------------|---------------------------------------------------------------|
| $\mathbf{X} \in \mathbb{R}^{N \times D}$                  | Original attribute matrix                                     |
| $\mathbf{X}^I \in \mathbb{R}^{N^I \times D}$              | Incomplete attribute matrix                                   |
| $\mathbf{N} \in \mathbb{R}^{N^I \times D}$                | Gaussian noise matrix                                         |
| $\tilde{\mathbf{X}}^I \in \mathbb{R}^{N^I \times D}$      | Corrupted incomplete attribute matrix                         |
| $\mathbf{A} \in \mathbb{R}^{N \times N}$                  | Original adjacency matrix                                     |
| $\tilde{\mathbf{A}} \in \mathbb{R}^{N \times N}$          | Normalized adjacency matrix                                   |
| $\tilde{\mathbf{A}}^I \in \mathbb{R}^{N^I \times N^I}$    | Normalized adjacency matrix of $\mathcal{G}^{Sub}$            |
| $\mathbf{I} \in \mathbb{R}^{N \times N}$                  | Identical matrix                                              |
| $\mathbf{C} \in \mathbb{R}^{N^I \times N^I}$              | Structure-attribute correlation matrix                        |
| $\tilde{\mathbf{A}}^{Ir} \in \mathbb{R}^{N^I \times N^I}$ | $r$ -order normalized adjacency matrix of $\mathcal{G}^{Sub}$ |
| $\mathbf{H}_A^I \in \mathbb{R}^{N^I \times d}$            | Attribute embeddings of incomplete samples                    |
| $\mathbf{H}_S \in \mathbb{R}^{N \times d}$                | Structure embeddings                                          |
| $\mathbf{H}_S^M \in \mathbb{R}^{N^M \times d}$            | Structure embeddings of missing samples                       |
| $\mathbf{H}_I \in \mathbb{R}^{N \times d}$                | Initially imputed embeddings                                  |
| $\mathbf{R} \in \mathbb{R}^{N \times N}$                  | Affinity structure                                            |
| $\mathbf{H} \in \mathbb{R}^{N \times d}$                  | Imputed embeddings                                            |
| $\mathbf{H}^M \in \mathbb{R}^{N^M \times d}$              | Imputed embeddings of missing samples                         |
| $\tilde{\mathbf{H}} \in \mathbb{R}^{N \times d}$          | Sample-recomposed embeddings                                  |
| $\mathbf{S} \in \mathbb{R}^{N \times N}$                  | Normalized self-correlated matrix                             |
| $\mathbf{M} \in \mathbb{R}^{N^I \times D}$                | Element-wise indicator matrix                                 |
| $\hat{\mathbf{X}} \in \mathbb{R}^{N \times D}$            | Rebuilt attribute matrix                                      |
| $\hat{\mathbf{X}}^I \in \mathbb{R}^{N^I \times D}$        | Rebuilt attribute matrix of incomplete samples                |
| $\hat{\mathbf{A}} \in \mathbb{R}^{N \times N}$            | Rebuilt adjacency matrix                                      |

corrupted sub-graph and the structural graph are transferred into two architecture-identical yet decoupled graph encoders to learn low-dimensional representations. In the information extraction phase, a structure-attribute consistency constraint allows the intermediate structure-attribute representations of attribute-incomplete samples to negotiate with each other to refine incomplete attributes. After that, the ITR mechanism utilizes the structure embeddings of the attribute-missing samples as the embedding initialization, and then adaptively refines these initial values by aggregating the reliable information of attribute-incomplete samples according to an affinity structure. Finally, RITR conducts the structure and attribute reconstructions over the latent embeddings by jointly minimizing three objectives.

#### C. Sample-denoising Then Consistency-preserving

This work attempts to solve an under-explored yet more challenging hybrid-absent graph machine learning task, where attribute-incomplete and attribute-missing samples exist simultaneously within a graph. The critical technique extension of RITR against its conference version is the STC component, which aims to make the learned representations robust to the attribute-incomplete input pattern and discriminative to serve the subsequent attribute-missing imputation task.

Since the network could not be optimized over unknown values, we employ a sample-denoising learning approach to ease the network training and facilitate the robustness of the learned attribute-incomplete features. Specifically, we first randomly generate a Gaussian noise matrix  $\mathbf{N} \in \mathbb{R}^{N^I \times D}$  with iterations, and then assign it to the incomplete attribute matrix  $\mathbf{X}^I \in \mathbb{R}^{N^I \times D}$  as initial values. The resultant matrix is denoted as a corrupted incomplete attribute matrix  $\tilde{\mathbf{X}}^I \in \mathbb{R}^{N^I \times D}$ . In

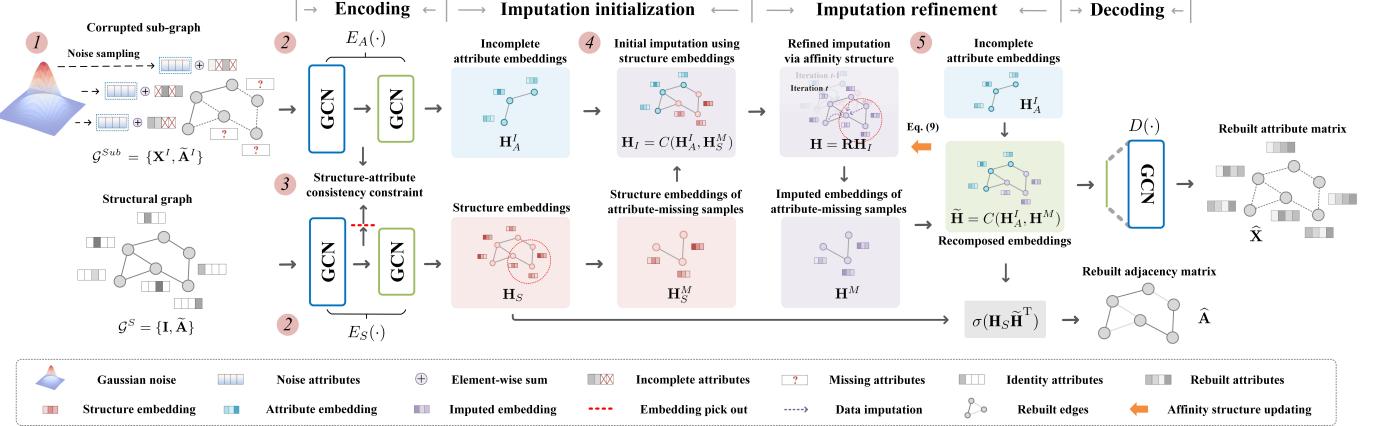


Fig. 2. The architecture of the revisiting initializing then refining (RITR) framework. To impute the incomplete values, we first initialize the original incomplete attributes as Gaussian noise for denoising learning (*i.e.*, step 1), and then introduce a structure-attribute consistency constraint to refine the incomplete values by approximating a structure-attribute correlation matrix to a high-order structural matrix (*i.e.*, step 3). To impute the missing values, we first adopt the structure embeddings of the attribute-missing samples as the embedding initialization (*i.e.*, step 4), and then adaptively refine these initial values by aggregating the reliable and informative information of the attribute-incomplete samples according to the affinity structure (*i.e.*, step 5).

the attribute-incomplete circumstance, we merely minimize the reconstruction errors of visible attributes between the original attribute matrix and the rebuilt attribute matrix for sample denoising. Although the sample-denoising scheme has been previously explored and proved to be powerful [44], directly applying it to the restoration of incomplete attributes is less comprehensive since partial attributes are invisible. To guarantee the high quality of representations of attribute-incomplete samples, it is intuitive that exploring the complete structure information could benefit the reconstruction of incomplete attributes since both types of latent variables share consistent and complementary properties of a typical graph [45]. Motivated by this, instead of directly addressing the incompleteness issue in the input space, we leverage the intimate relationship between structure-attribute embeddings to further refine the initial imputation of incomplete attributes. To be specific, we first utilize two graph convolution network-based encoders denoted as  $E_A(\cdot)$  and  $E_S(\cdot)$  to extract the latent features of attribute-incomplete samples and graph structure, respectively. Formally, given a sub-graph  $\tilde{\mathcal{G}}^{Sub}$  with a corrupted incomplete attribute matrix  $\tilde{\mathbf{X}}^I$  and a corresponding normalized adjacency matrix  $\tilde{\mathbf{A}}^I \in \mathbb{R}^{N^I \times N^I}$ ,  $E_A(\cdot)$  accepts them as input and the  $l$ -th latent representations of attribute-incomplete samples can be formulated as below:

$$\mathbf{H}_A^{I(l)} = \sigma(\tilde{\mathbf{A}}^I \mathbf{H}_A^{I(l-1)} \boldsymbol{\Theta}^{(l)}), \quad (1)$$

where  $\boldsymbol{\Theta}^{(l)}$  is the parameter matrix of  $E_A(\cdot)$  in the  $l$ -th layer and  $\sigma(\cdot)$  indicates a non-linear activation function. Similarly,  $E_S(\cdot)$  receives a structure graph  $\mathcal{G}^S$  with an identity matrix  $\mathbf{I} \in \mathbb{R}^{N \times N}$  and a normalized adjacency matrix  $\tilde{\mathbf{A}} \in \mathbb{R}^{N \times N}$ , and the  $l$ -th latent representations of the graph structure can be obtained:

$$\mathbf{H}_S^{(l)} = \sigma(\tilde{\mathbf{A}} \mathbf{H}_S^{(l-1)} \boldsymbol{\Psi}^{(l)}), \quad (2)$$

where  $\boldsymbol{\Psi}^{(l)}$  is the parameter matrix of  $E_S(\cdot)$  in the  $l$ -th layer. After that, we refine the intermediate information of attribute-incomplete samples via a structure-attribute consistency constraint, so as to make the learned representations much better

suit for subsequent attribute-missing imputation tasks. Concretely, we encourage each attribute-incomplete sample to be closer to its counterpart as well as  $r$ -order neighbors across structure-attribute modalities, which can be formulated as:

$$\mathcal{L}_C = \underbrace{\frac{1}{N^I} \sum_i^{N^I} (\mathbf{C}_{ii} - 1)^2}_{\text{self-loop consistency}} + \underbrace{\frac{1}{N^I(N^I - 1)} \sum_i^{N^I} \sum_{j \neq i}^{N^I} (\mathbf{C}_{ij} - \tilde{\mathbf{A}}_{ij}^{Ir})^2}_{\text{high-order structural consistency}}, \quad (3)$$

$$\mathbf{C} = \frac{\mathbf{H}_A^{I(1)} p(\mathbf{H}_S^{(1)})^T}{\|\mathbf{H}_A^{I(1)}\| \|p(\mathbf{H}_S^{(1)})\|}, \quad (4)$$

where  $\mathbf{C} \in \mathbb{R}^{N^I \times N^I}$  and  $\tilde{\mathbf{A}}^{Ir} \in \mathbb{R}^{N^I \times N^I}$  denote a structure-attribute correlation matrix and a  $r$ -order normalized adjacency matrix, respectively.  $N^I$  is the number of attribute-incomplete samples. In addition,  $\mathbf{H}_A^{I(1)}$  and  $\mathbf{H}_S^{(1)}$  indicate the embeddings of attribute-incomplete samples and graph structure in the first layer, respectively.  $p(\cdot)$  is an embedding pick-out function.

As seen in Eq.(3), the first term makes diagonal elements of the structure-attribute correlation matrix close to one value, causing the structure-attribute embeddings of each attribute-incomplete sample to be consistent. Moreover, the neighbors of each sample contain rich complementary information that should be considered for incomplete attribute completion. Hence, the second term encourages each sample to be closer to  $r$ -order neighbors than non-neighbors across structure-attribute modalities, aiming to exploit diverse complete structure properties to assist the feature completion of attribute-incomplete samples. By doing this, both sample denoising and structure-attribute consistency constraint are seamlessly integrated to 1) make the learned representations of attribute-incomplete samples robust and invariant to data perturbations (*e.g.*, the noise and incompleteness of the graph); 2) preserve more trustworthy features from available information to achieve better data imputation. The overall pipeline of training the proposed STC is summarized in Algorithm 1.

**Algorithm 1** The learning procedure of the STC mechanism

**Input:** Sub-graph  $\mathcal{G}^{Sub} = \{\mathbf{X}^I, \tilde{\mathbf{A}}^I\}$ ; structural graph  $\mathcal{G}^S = \{\mathbf{I}, \tilde{\mathbf{A}}\}$ ;  $r$ -order normalized adjacency matrix  $\tilde{\mathbf{A}}^{Ir}$  of  $\mathcal{G}^{Sub}$ ; pre-training iterations  $T$ ; hyper-parameters  $\beta$ .  
**Output:** Attribute embeddings of incomplete samples  $\mathbf{H}_A^I$  and structure embeddings  $\mathbf{H}_S$ .

---

```

1: for  $t = 1$  to  $T$  do
2:   Generate a Gaussian noise matrix and assign it to  $\mathbf{X}^I$ ;
3:   Utilize  $E_A(\cdot)$  to learn  $\mathbf{H}_A^I$  by Eq.(1);
4:   Utilize  $E_S(\cdot)$  to learn  $\mathbf{H}_S$  by Eq.(2);
5:   Calculate the structure-attribute correlation matrix by Eq.(4)
6:   Optimize the model with Adam by minimizing Eq.(3) with  $\beta$ ;
7: end for
8: return  $\mathbf{H}_A^I$  and  $\mathbf{H}_S$ 

```

---

#### D. Initializing Then Refining

**Imputation Initialization.** Besides completing and fine-tuning the samples with incomplete attributes, assigning initial values to the samples with missing samples should be further taken into account. A widely-adopted measure is traditional imputation techniques, such as zero value filling and mean value filling. Nevertheless, in the attribute-missing circumstance, these filling methods could incorporate amounts of irrelevant noise that will diffuse through the network, causing semantically biased representations. To alleviate this issue, it is intuitive to leverage the structure embeddings as the embedding initialization for latent variables of attribute-missing samples. The reason for that is two-fold. Firstly, the attribute embedding and the structure embedding describe different aspects of a node, providing consistent and complementary information in these two modalities [45]. Secondly, this initialization approach is reliable since the structure information of the original graph is complete.

To this end, after encoding incomplete attribute embeddings  $\mathbf{H}_A^I \in \mathbb{R}^{N^I \times d}$  and structure embeddings  $\mathbf{H}_S \in \mathbb{R}^{N \times d}$ , we first pick out the structure embeddings of attribute-missing samples  $\mathbf{H}_S^M \in \mathbb{R}^{N^M \times d}$  from  $\mathbf{H}_S$ , and then utilize a Concat function  $C(\cdot)$  to integrate  $\mathbf{H}_S^M$  with  $\mathbf{H}_A^I$ , where  $d$  refers to the embedding dimension. It is worth noting that the information concatenation to construct  $\mathbf{H}_I \in \mathbb{R}^{N \times d}$  is not the classic channel-wise or row-wise concatenation. In this operation, the latent variables of attribute-incomplete samples are filled with  $\mathbf{H}_A^I$  and the latent variables of attribute-missing samples are filled with  $\mathbf{H}_S^M$ :

$$\mathbf{H}_I = C(\mathbf{H}_A^I, \mathbf{H}_S^M), \quad (5)$$

where  $\mathbf{H}_I$  indicates the initially imputed embeddings. In our concatenation settings, the location of each sample remains unchanged within the original graph.

**Imputation Refinement.** It is well known that node attributes preserve the semantic graph content while the graph structure implies the connection relationships among samples. Consequently, the trustworthiness degrees of two-source information exhibit differences to some extent. Making full use of the trustworthy visible structure and attribute information to initial and refine the missing values may benefit the data imputation quality of attribute-missing samples. To illustrate whether our argument holds, we make a comparison among three methods

TABLE II  
PERFORMANCE COMPARISON. BOTH ATTRIBUTE-INCOMPLETE AND ATTRIBUTE-MISSING RATIOS ARE SET TO 60%.  $\uparrow$  DENOTES THE PERFORMANCE IMPROVEMENT OF OURS-S-A AGAINST OURS-S. THE BOLDFACE AND UNDERLINE VALUES INDICATE THE BEST AND THE SUB-OPTIMAL RESULTS (%), RESPECTIVELY.

| Dataset  | Method   | Recall@50                                 | NDCG@50                                   |
|----------|----------|-------------------------------------------|-------------------------------------------|
| Cora     | Ours-Z   | 27.85                                     | 25.68                                     |
|          | Ours-S   | <u>28.59</u>                              | <u>26.45</u>                              |
|          | Ours-S-A | <b>32.43 (3.84 <math>\uparrow</math>)</b> | <b>30.88 (4.43 <math>\uparrow</math>)</b> |
| Citeseer | Ours-Z   | 18.56                                     | 18.56                                     |
|          | Ours-S   | <u>19.79</u>                              | <u>19.38</u>                              |
|          | Ours-S-A | <b>22.14 (2.35 <math>\uparrow</math>)</b> | <b>23.67 (4.29 <math>\uparrow</math>)</b> |

and discuss the performance on Cora and Citeseer. Here we set both attribute-incomplete and attribute-missing ratios as 60%. Ours-Z, Ours-S, and Ours-S-A are methods where we impute the latent variables of attribute-missing samples with zero values, the structure embeddings merely, and the structure-attribute embeddings, respectively. As shown in Table II, we can observe that 1) Ours-S performs better than Ours-Z, which empirically verifies that the structure embeddings  $\mathbf{H}_S^M$  could provide an effective embedding initialization; 2) the trustworthy attributes adding in data imputation could provide more discriminative information to assist in the refinement of initial imputation. Specifically, we leverage available attribute properties  $\mathbf{H}_A^I$  to refine the initially imputed variables  $\mathbf{H}_S^M$  via an affinity structure  $\mathbf{R} \in \mathbb{R}^{N \times N}$ . The intuition behind this is that  $\mathbf{H}_A^I$  is trustworthy since all attribute-incomplete samples are carefully restored via the STC mechanism. By doing this, the semantic gap between  $\mathbf{H}_A^I$  and  $\mathbf{H}_S^M$  is allowed to be narrowed, thus boosting the discriminative capacity of the overall graph embedding. The above procedure can be written as a graph convolution-like formulation:

$$\mathbf{H} = \mathbf{R}\mathbf{H}_I, \quad (6)$$

where  $\mathbf{H} \in \mathbb{R}^{N \times d}$  indicates the imputed embeddings and we initialize the affinity structure  $\mathbf{R}$  as  $\tilde{\mathbf{A}}$ .

According to Eq.(6), we refine the attribute-missing imputation from the following two aspects. On the one hand, it is obvious that the noise information in  $\mathbf{H}_S^M$  can be transferred into the well-learned attribute embeddings of attribute-incomplete samples. This would undermine the representation quality and the reconstruction accuracy of available information, which in turn negatively affects the subsequent data imputation tasks and even distort the original graph. To tackle this problem, we implement an information recomposing scheme to decrease the adverse effect of noise information passing from the embeddings of attribute-missing samples. Firstly, we pick out the latent variables of attribute-missing samples  $\mathbf{H}^M \in \mathbb{R}^{N^M \times d}$  from  $\mathbf{H}$ , and next recompose them with  $\mathbf{H}_A^I$  using a Concat function  $C(\cdot)$ :

$$\tilde{\mathbf{H}} = C(\mathbf{H}_A^I, \mathbf{H}^M), \quad (7)$$

where  $\tilde{\mathbf{H}} \in \mathbb{R}^{N \times d}$  indicates the sample-recomposed embeddings. The information recomposing scheme replaces the adjusted embeddings of attribute-incomplete samples with more reliable  $\mathbf{H}_A^I$ . Meanwhile, as illustrated in Fig. 2, we fix the embeddings of attribute-incomplete samples as  $\mathbf{H}_A^I$  in the

**Algorithm 2** The learning procedure of the ITR mechanism

**Input:** Attribute embeddings of incomplete samples  $\mathbf{H}_A^I$ ; structure embeddings  $\mathbf{H}_S$ ; iterations  $T$ ; update interval  $U$ ; element-wise indicator matrix  $\mathbf{M}$ ; hyper-parameters  $\gamma$  and  $\alpha$ .  
**Output:** Rebuilt attribute matrix  $\widehat{\mathbf{X}}$  and adjacency matrix  $\widehat{\mathbf{A}}$ .

---

```

1: for  $t = 1$  to  $T$  do
2:   Obtain  $\mathbf{H}_S^M$  from  $\mathbf{H}_S$  in accordance with indices;
3:   Combine  $\mathbf{H}_A^I$  with  $\mathbf{H}_S^M$  for initial imputation by Eq.(5);
4:   Conduct the imputation refinement using  $\mathbf{R}$  by Eq.(6);
5:   Recompose the latent variables to obtain  $\widetilde{\mathbf{H}}$  by Eq.(7);
6:   Calculate  $\mathbf{S}$  over the resultant  $\widetilde{\mathbf{H}}$  by Eq.(8);
7:   if  $t\%U == 0$  then
8:     Update  $\mathbf{R}$  to refine  $\mathbf{H}$  by Eq.(9) and Eq.(6);
9:   end if
10:  Utilize  $D(\cdot)$  to decode  $\widetilde{\mathbf{H}}$  and output  $\widehat{\mathbf{X}}$  by Eq.(10);
11:  Utilize a simple inner decoder to rebuild  $\widehat{\mathbf{A}}$ ;
12:  Optimize the model with Adam by minimizing Eq.(11) and
    Eq.(12) with  $\alpha$ ;
13: end for
14: return  $\widehat{\mathbf{X}}$  and  $\widehat{\mathbf{A}}$ 

```

---

final step. This provides the most trustworthy information on attribute-incomplete samples for subsequent missing attribute imputation.

On the other hand, we argue that the initial affinity matrix  $\mathbf{R}$  (*i.e.*,  $\widetilde{\mathbf{A}}$ ) is not the ground truth. The limitations within this matrix are two-fold: 1) noisy connections. Besides inner connections within clusters, inappropriate connections could exist between clusters in the matrix; 2) missing connections. In  $\widetilde{\mathbf{A}}$ , only the first-order connections are preserved, the high-order relevant connections could be missing. Both would cause inaccurate imputation and reconstruction of missing attributes. To overcome these issues, we seek to refine  $\mathbf{R}$  by emphasizing the dependable connections while weakening the unreliable ones. To this end, we propose an affinity structure updating scheme to optimize  $\mathbf{R}$  with iterations. Specifically, we first calculate a normalized self-correlated matrix  $\mathbf{S} \in \mathbb{R}^{N \times N}$  according to  $\widetilde{\mathbf{H}}$  as below:

$$\mathbf{S}_{jk} = \mathcal{N}\left(\frac{\widetilde{\mathbf{h}}_j \widetilde{\mathbf{h}}_k^\top}{\|\widetilde{\mathbf{h}}_j\| \|\widetilde{\mathbf{h}}_k\|}\right), \quad \forall j, k \in [1, N], \quad (8)$$

where  $\mathcal{N}(\mathbf{Y}) = \mathbf{D}_Y^{-\frac{1}{2}} \mathbf{Y} \mathbf{D}_Y^{-\frac{1}{2}}$  indicates a structural normalization function,  $\mathbf{D}_Y \in \mathbb{R}^{N \times N}$  is the degree matrix of  $\mathbf{Y}$ .  $\widetilde{\mathbf{h}}_j$  ( $\widetilde{\mathbf{h}}_k$ ) indicates the embedding of sample  $\mathbf{v}_j$  ( $\mathbf{v}_k$ ). Then we optimize the affinity structure  $\mathbf{R}$  every  $t$  iterations via Eq.(9), and leverage it as guidance for the subsequent missing attribute imputation:

$$\mathbf{R} = \gamma \widetilde{\mathbf{A}} + (1 - \gamma) \mathbf{S}, \quad (9)$$

where  $\gamma$  is a balanced hyper-parameter and is initialized as 0.5. With the affinity structure updating scheme, the network is enabled to construct the embeddings of attribute-missing samples with not only the first-order but also the high-order connections within the graph structure. Since the embeddings of attribute-incomplete samples become more reliable and the embeddings of attribute-missing samples become more informative, the quality of data imputation could be further improved, making the learned representations more discriminative and robust. The overall pipeline of training the proposed ITR is summarized in Algorithm 2.

### E. Training Objectives and Complexity Analysis

1) *Training Objectives*: After obtaining  $\widetilde{\mathbf{H}}$ , we feed it with  $\widetilde{\mathbf{A}}$  into a graph decoder  $D(\cdot)$  to rebuilt the attributes of attribute-incomplete and attribute-missing samples:

$$\widetilde{\mathbf{H}}^{(l)} = \sigma(\widetilde{\mathbf{A}} \widetilde{\mathbf{H}}^{(l-1)} \Phi^{(l)}), \quad (10)$$

where  $\widetilde{\mathbf{W}}^{(l)}$  indicates the parameter matrix of  $D(\cdot)$  in the  $l$ -th layer.  $\widetilde{\mathbf{H}}^{(0)}$  and  $\widetilde{\mathbf{H}}^{(2)}$  denote the sample-recomposed embeddings  $\widetilde{\mathbf{H}}$  and the rebuilt attribute matrix  $\widehat{\mathbf{X}} \in \mathbb{R}^{N \times D}$ , respectively. The joint loss function of RITR includes three parts, which can be written as:

$$\mathcal{L}_A = \frac{1}{2N^I} \|\mathbf{M} \odot (\mathbf{X}^I - \widehat{\mathbf{X}}^I)\|_F^2, \quad (11)$$

$$\mathcal{L}_S = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N BCE(\widetilde{\mathbf{A}}_{ij}, \widehat{\mathbf{A}}_{ij}), \quad (12)$$

$$\mathcal{L} = \alpha \mathcal{L}_A + \mathcal{L}_S + \beta \mathcal{L}_C. \quad (13)$$

In Eq.(11),  $\mathcal{L}_A$  refers to the mean square error (MSE) of attribute-incomplete samples between  $\mathbf{X}$  and  $\widehat{\mathbf{X}}$ .  $\mathbf{M} \in \mathbb{R}^{N^I \times D}$  is an element-wise indicator matrix where  $M_{ij} = 1$  if  $X_{ij}^I$  is a real value, otherwise  $X_{ij}^I$  is a null value (*i.e.*, an incomplete attribute). In Eq.(12),  $\mathcal{L}_S$  refers to the binary cross-entropy (BCE) between the normalized adjacency matrix  $\widetilde{\mathbf{A}}$  and the rebuilt adjacency matrix  $\widehat{\mathbf{A}} \in \mathbb{R}^{N \times N}$ , where  $\widehat{\mathbf{A}} = \sigma(\mathbf{H}_S \widetilde{\mathbf{H}}^\top)$ ,  $\sigma(\cdot)$  is a Sigmoid activation function.  $\alpha$  and  $\beta$  are two balanced hyper-parameters. The applied optimization objectives are similar to existing attribute-missing graph machine learning methods [14], [17]. However, the major differences between current methods and our improved method could be summarized in the following three parts: 1) more naturally handling hybrid-absent graphs in an unsupervised circumstance; 2) more comprehensive that seamlessly unifies the representation learning and data imputation processes of attribute-incomplete and attribute-missing samples into a common framework; 3) more discriminative that enables the structure-attribute information to sufficiently negotiate with each other for feature completion by performing STC and ITR mechanisms.

2) *Complexity Analysis*: The time complexity of the proposed RITR could be discussed from the following two aspects: the graph auto-encoder framework and the loss error computation. For two GCN-based graph encoders, the complexities of  $E_A(\cdot)$  and  $E_s(\cdot)$  are  $\mathcal{O}(Nd^2(L-1) + NdD_a + |\mathcal{E}|dL)$  and  $\mathcal{O}(Nd^2(L-1) + NdD_s + |\mathcal{E}|dL)$ , where  $N$ ,  $L$ ,  $|\mathcal{E}|$  are the number of nodes, encoder layers, and edges, respectively.  $D_a$ ,  $D_s$ , and  $d$  are the dimension of raw attribute features, raw structure features, and hidden representations, respectively. For the graph decoder, the complexity of  $D(\cdot)$  is  $\mathcal{O}(Nd^2(L-1) + NdD_a + |\mathcal{E}|d(L-1) + |\mathcal{E}|D_a)$ . For the loss error computation, we follow SAT [14] and use the MSE and BCE loss functions to reconstruct node attributes and graph structure, respectively. The time complexities of  $\mathcal{L}_A$  and  $\mathcal{L}_S$  are  $\mathcal{O}(nD_a)$  and  $\mathcal{O}(N^2)$ . The time complexity of the structure-attribute consistency loss function  $\mathcal{L}_C$  is  $\mathcal{O}(n^2)$ , where  $n = (1 - \theta)N$ ,  $\theta$  is the ratio of attribute-missing samples. Considering the attribute-missing problem is universal in

TABLE III  
A COMPARISON OF THE COMPLEXITY ANALYSIS AMONG SAT [14], ITR [17], SVGA [42], AND THE PROPOSED RITR.

| Model       | Time complexity (model and loss error computation)                                                      |
|-------------|---------------------------------------------------------------------------------------------------------|
| SAT         | $\mathcal{O}(Nd(D_a + D_s + dL) +  \mathcal{E} dL + N^2 + nD_a) \approx \mathcal{O}(N^2)$               |
| SVGA        | $\mathcal{O}(Nd^2L +  \mathcal{E} dL + NdD_a + Ndc + ND_a + N + N^2) \approx \mathcal{O}(N^2)$          |
| ITR         | $\mathcal{O}(Nd(D_a + D_s + dL) +  \mathcal{E} (dL + D_a) + N^2 + nD_a) \approx \mathcal{O}(N^2)$       |
| RITR (Ours) | $\mathcal{O}(Nd(D_a + D_s + dL) +  \mathcal{E} (dL + D_a) + N^2 + n(n + D_a)) \approx \mathcal{O}(N^2)$ |

real-world scenarios (*i.e.*,  $\theta$  can be set to a large value) and  $n^2$  can be a relatively small value, the computation overhead here is acceptable. The overall time complexity of RITR for each training iteration is  $\mathcal{O}(Nd(D_a + D_s + dL) + |\mathcal{E}|(dL + D_a) + N^2 + n(n + D_a)) \approx \mathcal{O}(N^2)$ . For a fair comparison, we conduct a complexity comparison among four attribute-missing graph machine learning methods and report the results in Table III. As seen, RITR consistently outperforms all baselines in both attribute-missing and hybrid-absent cases (see Sections V-B and V-C), requiring no additional computation complexity compared to these competitors.

#### IV. EXPERIMENTS

In this section, we evaluate the effectiveness of ITR and RITR against some advanced graph machine learning methods. The experiments aim to answer the following five questions:

- **Q1.** How do the proposed methods perform compared to baselines in profiling and node classification tasks?
- **Q2.** How does the designed component influence the performance?
- **Q3.** How does the proposed method perform with different absent ratios?
- **Q4.** How do key hyper-parameters influence the performance of the proposed method?
- **Q5.** How about the method convergence and performance variation with iterations?

In the following, we begin with a brief experimental setup introduction, including benchmark datasets, implementation procedures, training settings, and compared methods. Then we report experimental results with corresponding analysis.

##### A. Experimental Setup

1) *Benchmark Datasets:* We implement experiments to evaluate two proposed methods, *i.e.*, ITR and RITR on four benchmark datasets, including Cora, Citeseer, Amazon Computer, and Amazon Photo. We summarize the detailed dataset information in Table IV.

- Cora<sup>1</sup> and Citeseer<sup>1</sup> are two popular citation network datasets. Specially, nodes mean scientific publications, and edges mean citation relationships. Each node has a predefined feature with corresponding dimensions.
- Amazon Photo<sup>2</sup> and Amazon Computers<sup>2</sup> (Amap and Amac for abbreviation) are segments of the Amazon co-purchase network, where nodes represent goods, edges indicate that two goods are frequently bought together, node features are bag-of-words encoded product reviews, and class labels are given by the product category.

<sup>1</sup><https://docs.dgl.ai/api/python/dgl.data.html#citation-network-dataset>

TABLE IV  
SUMMARY OF DATASETS.

| Dataset  | Nodes | Edges  | Dimension | Classes |
|----------|-------|--------|-----------|---------|
| Cora     | 2708  | 5278   | 1433      | 7       |
| Citeseer | 3327  | 4228   | 3703      | 6       |
| Amac     | 13752 | 245861 | 767       | 10      |
| Amap     | 7650  | 119081 | 745       | 8       |

2) *Implementation Procedures:* Both ITR and RITR are implemented with the Pytorch platform. We evaluate the effectiveness of two proposed methods through a two-step learning procedure. Firstly, we train an unsupervised framework to learn node representations and complete absent information for at least 600 iterations. Following SAT [14], we regard the profiling learning as a pretext task and adopt Recall@K and NDCG@K as metrics to evaluate the quality of rebuilt attributes. To alleviate the over-fitting problem, we perform an early stop strategy when the loss value reaches a plateau. Secondly, for the node classification task, we feed the rebuilt attribute matrix into a graph classifier and optimize it with five-fold validation 10 times, and report the average accuracy (ACC) performance.

3) *Training Settings:* In the attribute-missing case, we record the performance of all methods directly according to the paper of SAT [14] except for GINN [10], GCNMF [11], and SVGA [42]. In the hybrid-absent case, we run the released source code of all compared methods by following the settings of the corresponding literature, and report their results. For our proposed ITR and RITR, we strictly follow the criterion of data splits as was done in SAT, including the split ratio of attribute-complete/missing samples and the split ratio of train/test sets. Specifically, 1) in the profiling task, we randomly sample 40% nodes with complete attributes as the training set, and manually mask all attributes of the rest of 10% and 50% nodes (*i.e.*, attribute-missing samples) as the validation set and the test set, respectively. Besides, when attribute-incomplete and attribute-missing samples exist simultaneously within a graph, we randomly mask 60% attributes of each attribute-complete sample (*i.e.*, the training set) before network learning. We employ a 4-layer graph auto-encoder framework and optimize it with Adam optimization algorithm. During the training phase, we transfer all samples into ITR and RITR to complete absent attributes by merely reconstructing the available ones. After training, we rebuild the attribute matrix over the well-trained model via forwarding propagation; 2) in the node classification task, we randomly split the rebuilt attributes into 80% and 20% for training and testing, respectively. We train the classifier with five-fold validation for 1000 iterations and repeat the experiments 10 times. According to the results

TABLE V  
PROFILING PERFORMANCE COMPARISON IN THE ATTRIBUTE-MISSING CIRCUMSTANCE. THE ATTRIBUTE-MISSING RATIO IS SET TO 60%. THE BOLDFACE AND UNDERLINE VALUES INDICATE THE BEST AND THE SUB-OPTIMAL RESULTS (%), RESPECTIVELY.

| Dataset  | Metric    | NeighAggre | VAE   | GCN   | GraphSage | GAT   | Hers  | GraphRNA | ARWMF | SVGA  | SAT   | ITR          | RITR         |
|----------|-----------|------------|-------|-------|-----------|-------|-------|----------|-------|-------|-------|--------------|--------------|
| Cora     | Recall@10 | 9.06       | 8.87  | 12.71 | 12.84     | 13.50 | 12.26 | 13.95    | 12.91 | 16.06 | 15.08 | <u>16.82</u> | <b>17.32</b> |
|          | Recall@20 | 14.13      | 12.28 | 17.72 | 17.84     | 18.12 | 17.23 | 20.43    | 18.13 | 23.17 | 21.82 | <u>23.69</u> | <b>24.47</b> |
|          | Recall@50 | 19.61      | 21.16 | 29.62 | 29.72     | 29.72 | 27.99 | 31.42    | 29.60 | 35.83 | 34.29 | <u>36.47</u> | <b>36.81</b> |
|          | NDCG@10   | 12.17      | 12.24 | 17.36 | 17.68     | 17.91 | 16.94 | 19.34    | 18.24 | 23.19 | 21.12 | <u>23.20</u> | <b>24.20</b> |
|          | NDCG@20   | 15.48      | 14.52 | 20.76 | 21.02     | 20.99 | 20.31 | 23.62    | 21.82 | 26.81 | 25.46 | <u>27.81</u> | <b>28.97</b> |
|          | NDCG@50   | 18.50      | 19.24 | 27.02 | 27.28     | 27.11 | 25.96 | 29.38    | 27.76 | 33.71 | 32.12 | <u>34.60</u> | <b>35.52</b> |
| Citeseer | Recall@10 | 5.11       | 3.82  | 6.20  | 6.12      | 5.61  | 5.76  | 7.77     | 5.52  | 8.83  | 7.64  | <u>9.63</u>  | <b>10.16</b> |
|          | Recall@20 | 9.08       | 6.68  | 10.97 | 10.97     | 10.12 | 10.25 | 12.72    | 10.15 | 14.55 | 12.80 | <u>15.48</u> | <b>15.97</b> |
|          | Recall@50 | 15.01      | 12.96 | 20.52 | 20.58     | 19.57 | 19.73 | 22.71    | 19.52 | 25.91 | 23.77 | <u>26.84</u> | <b>26.90</b> |
|          | NDCG@10   | 8.23       | 6.01  | 10.26 | 10.03     | 8.78  | 9.04  | 12.91    | 8.59  | 15.21 | 12.98 | <u>16.32</u> | <b>17.16</b> |
|          | NDCG@20   | 11.55      | 8.39  | 14.23 | 13.93     | 12.53 | 12.79 | 17.03    | 12.45 | 20.08 | 17.29 | <u>21.20</u> | <b>22.03</b> |
|          | NDCG@50   | 15.60      | 12.51 | 20.49 | 20.34     | 18.72 | 19.00 | 23.58    | 18.58 | 27.99 | 24.47 | <u>28.69</u> | <b>29.22</b> |
| Amac     | Recall@10 | 3.21       | 2.55  | 2.73  | 2.69      | 2.71  | 2.73  | 3.86     | 2.80  | 3.97  | 3.91  | <u>4.48</u>  | <b>4.59</b>  |
|          | Recall@20 | 5.93       | 5.02  | 5.33  | 5.28      | 5.30  | 5.25  | 6.90     | 5.44  | 7.22  | 7.03  | <u>7.81</u>  | <b>8.06</b>  |
|          | Recall@50 | 13.06      | 11.96 | 12.75 | 12.78     | 12.78 | 12.73 | 14.65    | 12.89 | 15.75 | 15.14 | <u>16.20</u> | <b>16.48</b> |
|          | NDCG@10   | 7.88       | 6.32  | 6.71  | 6.64      | 6.73  | 6.76  | 9.31     | 6.94  | 10.11 | 9.63  | <u>10.95</u> | <b>11.14</b> |
|          | NDCG@20   | 11.56      | 9.70  | 10.27 | 10.20     | 10.28 | 10.25 | 13.33    | 10.53 | 14.98 | 13.79 | <u>15.37</u> | <b>15.69</b> |
|          | NDCG@50   | 19.23      | 17.21 | 18.24 | 18.22     | 18.30 | 18.25 | 21.55    | 18.51 | 23.84 | 22.43 | <u>24.29</u> | <b>24.63</b> |
| Amap     | Recall@10 | 3.29       | 2.76  | 2.94  | 2.95      | 294   | 2.92  | 3.90     | 2.94  | 4.14  | 4.10  | <u>4.36</u>  | <b>4.49</b>  |
|          | Recall@20 | 6.16       | 5.38  | 5.73  | 5.62      | 5.73  | 5.74  | 7.03     | 5.68  | 7.51  | 7.43  | <u>7.82</u>  | <b>8.00</b>  |
|          | Recall@50 | 13.61      | 12.79 | 13.24 | 13.22     | 13.24 | 13.28 | 15.08    | 13.27 | 15.93 | 15.97 | <u>16.43</u> | <b>16.61</b> |
|          | NDCG@10   | 8.13       | 6.75  | 7.05  | 7.12      | 7.05  | 7.14  | 9.59     | 7.27  | 10.21 | 10.06 | <u>10.73</u> | <b>10.94</b> |
|          | NDCG@20   | 11.96      | 10.31 | 10.82 | 10.79     | 10.83 | 10.94 | 13.77    | 10.98 | 14.73 | 14.50 | <u>15.33</u> | <b>15.56</b> |
|          | NDCG@50   | 19.98      | 18.30 | 18.93 | 18.96     | 18.92 | 19.06 | 22.32    | 19.15 | 23.75 | 23.59 | <u>24.50</u> | <b>24.72</b> |

of parameter sensitivity testing, we fix two balanced hyperparameters  $\alpha$  and  $\beta$  to 10. 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. Please note that we do not carefully tune these parameters for ease of training as was done in SAT.

4) *Compared Methods:* We compare RITR with the existing 13 baseline methods for future estimation on attribute-missing and hybrid-absent graphs. Specifically, **NeighAggre** (NAS' 08) [46] is a classical profiling method. **VAE** (NeurIPS' 16) [47] is a well-known auto-encoder method. **GCN** (ICLR' 17) [48], **GraphSage** (NeurIPS' 17) [49], and **GAT** (ICLR' 18) [16] are three typical graph neural networks. **GraphRNA** (KDD' 19) [50] and **ARWMF** (NeurIPS' 19) [51] are representatives of attributed random walk-based methods. **Hers** (AAAI' 19) [52] is a cold-start recommendation method. **SAT** (TPAMI' 22) [14] is the first attribute-missing graph imputation network. **SVGA** (KDD' 22) [42] and **ITR** (IJCAI' 22) [17] are two most advanced attribute-missing graph autoencoders. **GINN** (NN' 20) [10] and **GCNMF** [10] (FGCS' 21) [11] are two state-of-the-art attribute-incomplete graph machine learning methods.

### B. Attribute-missing Case

1) *Performance Comparison (Q1):* As shown in Table V, we report the profiling performance of all methods mentioned above. This table shows that ITR and RITR outperform all compared baseline methods in terms of six metrics on four datasets. Specifically, 1) we first compare NeighAggre and VAE with our methods. Instead of merely exploiting the structure or attribute information for data imputation, our methods have two-source information sufficiently negotiate with each other, thus consistently exceeding NeighAggre and VAE by a large margin; 2) ITR and RITR show superior performance against GCN, GraphSage, and GAT, all of which have demonstrated strong representation learning capability in handling

attribute-complete graphs. However, the results show that these methods are not suitable to solve the attribute-missing problem; 3) for the two strongest attribute-missing graph machine learning methods (*i.e.*, SVGA and SAT), RITR outperforms them by 1.81%/3.40%, 1.23%/4.75%, 0.79%/2.20%, and 0.97%/1.13% in terms of NDCG@50 metric on four datasets, respectively. This is because these baselines heavily rely on pre-defined assumptions that may not always hold in real-world graphs for data imputation, while RITR does not make any prior distribution assumption so that it can flexibly and effectively make full use of visible information for feature completion; 4) RITR further achieves better performance than ITR on all datasets. These superior results of RITR over the state-of-the-art method further verify the effectiveness of our improved framework for handling attribute-missing graphs.

Moreover, we report the node classification performance of 8 methods in Table VI. “X” or “X+A” indicates that the classifier receives the attribute matrix or attribute and adjacency matrices as input in the node classification task. Note that here we only take the attribute-missing case (*i.e.*, marked as “AM”) into consideration. From these results, we can see that 1) the classification results of GINN and GCNMF are not comparable to those of our two methods. ITR and RITR achieve at least 15.26%/15.51%, 4.69%/5.61%, 6.38%/7.22%, and 4.08%/4.45 accuracy increment. This indicates that these attribute-incomplete methods fall into inaccurate data imputation with extremely limited observations so that they can not learn effective representations; 2) taking the performance of “X” for instance, ITR and RITR gain 4.99%/5.20%, 7.05%/7.37%, 9.78%/11.18%, and 3.13%/3.65% performance enhancement over the state-of-the-art SAT method. Similar observations can be obtained among SVGA, ITR, and RITR. These benefits can be attributed to the following merits: 1) different from SVGA and SAT, our proposed graph imputation networks avoid the reliance on any prior distribution assumption for missing attribute completion, so that they can facilitate

TABLE VI

NODE CLASSIFICATION PERFORMANCE COMPARISON. “AM” REFERS TO THE ATTRIBUTE-MISSING CIRCUMSTANCE. “HA” REFERS TO THE HYBRID-ABSENT CIRCUMSTANCE. BOTH ATTRIBUTE-INCOMPLETE AND ATTRIBUTE-MISSING RATIOS ARE SET TO 60%. THE BOLDFACE AND UNDERLINE VALUES INDICATE THE BEST AND THE SUB-OPTIMAL RESULTS (%), RESPECTIVELY.

| Type | Dataset  | GCN   |       | GAT   |       | GINN  |       | GCNMF |       | SVGA  |       | SAT   |       | ITR          |              | RITR         |              |
|------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------------|--------------|--------------|--------------|
|      |          | AM    | HA    | AM           | HA           | AM           | HA           |
| X    | Cora     | 39.43 | 29.39 | 41.43 | 29.39 | -     | -     | -     | -     | 78.70 | 67.91 | 76.44 | 72.53 | <b>81.43</b> | <u>75.04</u> | <b>81.64</b> | <b>78.11</b> |
|      | Citeseer | 37.68 | 20.96 | 21.29 | 21.26 | -     | -     | -     | -     | 62.33 | 54.51 | 60.10 | 54.15 | <u>67.15</u> | <u>61.05</u> | <b>67.47</b> | <b>63.75</b> |
|      | Amac     | 36.60 | 35.76 | 37.47 | 36.98 | -     | -     | -     | -     | 72.56 | 64.44 | 74.10 | 68.94 | <u>83.88</u> | <u>80.68</u> | <b>85.28</b> | <b>82.92</b> |
|      | Amap     | 26.83 | 25.70 | 25.98 | 25.39 | -     | -     | -     | -     | 88.55 | 82.49 | 87.62 | 84.53 | <u>90.75</u> | <u>88.98</u> | <b>91.27</b> | <b>90.49</b> |
| X+A  | Cora     | 43.87 | 30.40 | 45.25 | 30.80 | 67.58 | 38.16 | 70.30 | 57.19 | 83.78 | 77.70 | 83.27 | 80.37 | <u>85.56</u> | <u>82.53</u> | <b>85.81</b> | <b>84.29</b> |
|      | Citeseer | 40.79 | 26.89 | 26.88 | 26.49 | 55.32 | 23.66 | 63.40 | 50.90 | 66.19 | 60.30 | 65.99 | 64.16 | <u>68.09</u> | <u>64.93</u> | <b>69.01</b> | <b>66.00</b> |
|      | Amac     | 39.74 | 38.88 | 40.34 | 39.78 | 81.27 | 25.45 | 76.43 | 35.40 | 85.87 | 80.56 | 85.19 | 83.33 | <u>87.65</u> | <u>84.96</u> | <b>88.49</b> | <b>86.19</b> |
|      | Amap     | 36.56 | 36.08 | 37.89 | 37.75 | 87.77 | 37.25 | 87.79 | 82.04 | 89.90 | 84.33 | 91.63 | 90.49 | <u>91.87</u> | <u>88.98</u> | <b>92.24</b> | <b>91.75</b> |

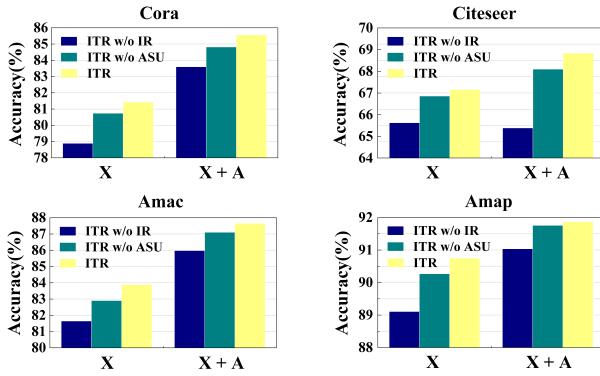


Fig. 3. Effect of the information recomposing (IR) and affinity structure updating (ASU) schemes for node classification.

the structure-attribute negotiation more flexibly and comprehensively; 2) the trustworthy visible attribute information and structure information can be used unitedly by ITR and RITR for data imputation instead of being treated separately. The above experimental results well demonstrate the superiority of ITR and RITR in the attribute-missing case.

2) *Effect of Two Schemes in ITR Criterion (Q2)*: To verify the benefit of the initializing-then-refining imputation criterion, we conduct ablation studies on four datasets to compare ITR and two ITR variants, each of which has one of the critical components removed. ITR w/o IR and ITR w/o ASU indicate the method with information recomposing and affinity structure updating being masked, respectively. From the results in Fig. 3, we observe that the accuracy of ITR on four datasets would degrade without one of the key components. Specifically, for the “X” task, ITR exceeds ITR w/o IR by 2.54%, 1.53%, 2.24%, and 1.64% accuracy increment, and ITR w/o ASU by 0.69%, 0.30%, 0.98%, and 0.55% accuracy increment on Cora, Citeseer, Amac, and Amap, respectively. We find that the information recomposing scheme plays a more important role than the information refining scheme. To visually illustrate this point, we present the mean square error comparison of ITR and ITR w/o IR at the last training iteration. As seen in Fig. 4, the method with the IR scheme achieves a better convergence than ITR w/o IR. This indicates that our information recomposing operation can effectively prohibit inaccurate information from being propagated, so the network can learn reliable representations for high-quality missing attribute restoration. All the above observations demonstrate the

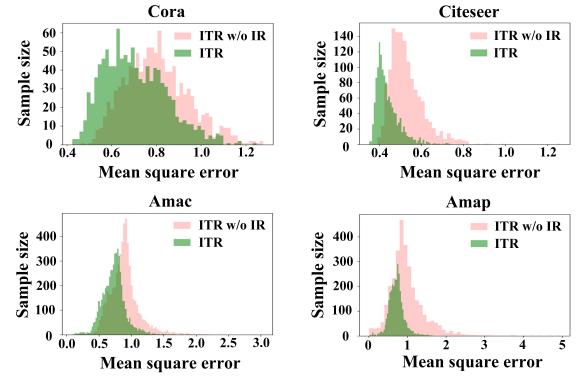


Fig. 4. MSE comparison of ITR and an ITR variant. ITR w/o IR denotes the ITR without information recomposing scheme.

effectiveness of our proposed ITR imputation criterion, which can enable the structure-attribute information to sufficiently negotiate with each other for more accurate data imputation.

### C. Hybrid-absent Case

1) *Performance Comparison (Q1)*: In this section, we further investigate the performance of our proposed methods and study a more challenging hybrid-absent (*i.e.*, marked as “HA”) problem that both attribute-missing and attribute-incomplete samples exist simultaneously within a graph. “X” or “X+A” indicates that the classifier receives the attribute matrix or attribute and adjacency matrices as input in the node classification task. To evaluate the quality of the rebuilt attribute matrix, we take six methods (*i.e.*, GCN, GAT, GINN, GCNMF, SVGA, and SAT) as baselines and report the classification accuracy on four datasets. From these results in Table VI, we can find that 1) although ITR outperforms all baseline methods and achieves the most competitive results, it suffers from a significant performance degradation by 4.37% on average in the “HA”(“X”) case compared to that in the “AM”(“X”) case. This is because ITR conducts sample embedding over attribute-incomplete samples directly so that amounts of error information have diffused through the network. As a result, the resultant representations are inaccurate and can hardly provide the attribute-missing samples with discriminative enough information for feature completion; 2) taking the “HA”(“X”) case for example, RITR achieves the best performance against all compared baselines. Specifically, RITR improves SVGA

TABLE VII

ABLATION STUDY ON THE EFFECTIVENESS OF EACH COMPONENT OF RITR. RITR w/o STC AND RITR w/o ITR INDICATE THAT THE METHOD WITHOUT THE SAMPLE-DENOISING THEN CONSISTENCY-PRESERVING COMPONENT AND THE INITIALIZING THEN REFINING COMPONENT, RESPECTIVELY. BOTH ATTRIBUTE-INCOMPLETE AND ATTRIBUTE-MISSING RATIOS ARE SET TO 60%. THE BOLDFACE AND UNDERLINE VALUES INDICATE THE BEST AND THE SUB-OPTIMAL RESULTS (%), RESPECTIVELY.

| Dataset  | Method       | Recall@10    | Recall@20    | Recall@50    | NDCG@10      | NDCG@20      | NDCG@50      |
|----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Cora     | RITR w/o STC | <u>13.63</u> | <u>19.67</u> | <u>30.91</u> | <u>19.45</u> | <u>23.47</u> | <u>29.43</u> |
|          | RITR w/o ITR | 12.03        | 16.52        | 27.70        | 16.79        | 19.84        | 25.66        |
|          | RITR         | <b>14.36</b> | <b>20.69</b> | <b>32.43</b> | <b>20.51</b> | <b>24.69</b> | <b>30.88</b> |
| Citeseer | RITR w/o STC | <u>7.08</u>  | <u>11.93</u> | <u>21.61</u> | <u>12.12</u> | <u>16.37</u> | <u>22.80</u> |
|          | RITR w/o ITR | 5.19         | 9.30         | 18.17        | 8.03         | 11.46        | 17.25        |
|          | RITR         | <b>7.78</b>  | <b>12.51</b> | <b>22.14</b> | <b>13.38</b> | <b>17.33</b> | <b>23.67</b> |
| Amac     | RITR w/o STC | <u>3.92</u>  | <u>6.97</u>  | <u>15.10</u> | <u>9.84</u>  | <u>14.05</u> | <u>22.72</u> |
|          | RITR w/o ITR | 2.83         | 5.49         | 12.92        | 7.01         | 10.63        | 18.60        |
|          | RITR         | <b>4.26</b>  | <b>7.47</b>  | <b>15.72</b> | <b>10.44</b> | <b>14.73</b> | <b>23.50</b> |
| Amap     | RITR w/o STC | <u>3.96</u>  | <u>7.16</u>  | <u>15.58</u> | <u>10.04</u> | <u>14.35</u> | <u>23.33</u> |
|          | RITR w/o ITR | 3.04         | 5.79         | 13.46        | 7.53         | 11.25        | 19.47        |
|          | RITR         | <b>4.25</b>  | <b>7.50</b>  | <b>16.00</b> | <b>10.42</b> | <b>14.78</b> | <b>23.84</b> |

and SAT by 10.20%/5.58%, 9.24%/9.60%, 18.48%/13.98%, and 8.00%/5.96% accuracy increment on all datasets. The observations of other cases are similar. These results once again verify that when both attribute-incomplete and attribute-missing samples exist simultaneously, the feature completion mechanisms of these baselines may have an adverse effect on the quality of recovered attributes due to incorrect information diffusion. RITR effectively alleviates this adverse influence by designing two personalized feature completion components with the initializing-then-refining imputation criterion.

2) *Effect of Each Component of RITR (Q2):* Here we conduct an ablation study to validate the effectiveness of our proposed attribute-incomplete and attribute-missing imputation mechanisms. Table VII reports the Recall and NDCG performance of three methods, including RITR w/o STC, RITR w/o ITR, and RITR. Specially, RITR w/o STC or RITR w/o ITR indicates that the method removes the sample-denoising then consistency-preserving component or the initializing then refining component. Note that here we set both attribute-missing and incomplete ratios as 60%. Table VII compares the results of RITR and its two variants, from which we can see that 1) RITR consistently improves RITR w/o STC on all datasets. Taking the results on Cora for instance, RITR gains 0.73%, 1.02%, 1.52%, 1.06%, 1.22%, and 1.45% increment in terms of Recall and NDCG, demonstrating the effectiveness of leveraging the intimate structure-attribute relationship to guide the imputation of incomplete attributes. Similar observations can be concluded from the results on other datasets; 2) RITR significantly outperforms RITR w/o ITR and has performance enhancements of 4.73%, 3.97%, 2.80%, and 2.54% over it in terms of Recall@50 on four datasets, respectively. These results imply the importance of an effective imputation strategy in which we employ the most trustworthy visible information to implement the missing attribute completion. In summary, this ablation study clearly validates that each component can contribute to the overall performance of RITR.

3) *Analysis of The Attribute-incomplete Ratio (Q3):* To further investigate the superiority of RITR, it is necessary to show whether the proposed RITR can still achieve effective feature completion when less visible attribute information is available. To this end, we make a performance comparison

between SAT and RITR by varying the attribute-incomplete ratio from 10% to 70% and fixing the attribute-missing ratio as 60%. From the results in Table VIII, several observations can be summarized: 1) RITR consistently performs better than SAT in all situations on four datasets. For example, RITR outperforms SAT by 2.42%, 2.25%, 1.85%, 1.84%, 1.83%, 1.94%, and 1.51% in terms of Recall@10 when the attribute-incomplete ratio varies from 10% to 70% on Citeseer. The observations of other metrics and datasets are similar. This is because SAT can not implement an effective latent distribution matching between incomplete attributes and structures. Naturally, the resultant misleading information poses a negative impact on data imputation and feature completion, resulting in sub-optimal node representations. RITR can effectively model hybrid-absent graphs and alleviate the diffusion of inaccurate information under the guidance of initializing-then-refining imputation criterion; 2) taking the results of Recall@10/NDCG@10 on Amac and Amap for example, RITR with 70% incomplete attributes can still achieve better performance than SAT with 10% ones. These results illustrate that RITR can still achieve high-quality data imputation and feature completion with limited observed signals. Overall, all the above results solidly demonstrate the superiority and robustness of RITR.

4) *Hyper-parameter Analysis (Q4):* As seen in Eq.(13), RITR 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  $\alpha$  and  $\beta$ . Note that we first set one to a certain value and then tune the other carefully. Fig. 5 reports the Recall and NDCG performance variation of RITR on Cora and Citeseer when  $\alpha$  and  $\beta$  vary from 1 to 20 with a step size of 5. From these sub-figures, we can observe that 1) tuning both  $\alpha$  and  $\beta$  would cause performance variation and the model performance is more stable in the range of [5, 15], suggesting that searching  $\alpha$  and  $\beta$  values from a reasonable hyper-parameter region could benefit the model performance; 2) for a certain  $\alpha$  value, the performance shows a trend of first rising and then dropping slightly with the variation of  $\beta$ . This indicates that RITR needs a proper coefficient to guarantee the structure-attribute consistency for improving the quality of feature completion.

TABLE VIII

PERFORMANCE COMPARISON BETWEEN SAT AND OUR PROPOSED RITR. WE FIX THE ATTRIBUTE-MISSING RATIO AS 60% AND VARY THE ATTRIBUTE-INCOMPLETE RATIO (AIR) FROM 10% TO 70%. THE BOLDFACE VALUES INDICATE THE BEST RESULTS (%).

| Dateset   | Cora     |              |       |              |       |              |       |              |       |              |       |              |       |              |
|-----------|----------|--------------|-------|--------------|-------|--------------|-------|--------------|-------|--------------|-------|--------------|-------|--------------|
| AIR       | 10%      |              | 20%   |              | 30%   |              | 40%   |              | 50%   |              | 60%   |              | 70%   |              |
| Method    | SAT      | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         |
| Recall@10 | 14.74    | <b>17.12</b> | 14.84 | <b>16.59</b> | 14.23 | <b>16.08</b> | 13.84 | <b>15.86</b> | 13.59 | <b>14.95</b> | 12.62 | <b>14.36</b> | 12.34 | <b>13.16</b> |
| Recall@20 | 21.45    | <b>23.72</b> | 21.46 | <b>23.36</b> | 20.70 | <b>23.02</b> | 20.17 | <b>22.29</b> | 19.57 | <b>21.40</b> | 18.46 | <b>20.69</b> | 16.78 | <b>18.72</b> |
| Recall@50 | 34.08    | <b>36.32</b> | 34.00 | <b>35.76</b> | 33.06 | <b>35.19</b> | 32.34 | <b>34.38</b> | 31.48 | <b>33.38</b> | 29.97 | <b>32.43</b> | 27.75 | <b>30.21</b> |
| NDCG@10   | 20.70    | <b>23.94</b> | 20.73 | <b>23.35</b> | 20.12 | <b>22.66</b> | 19.50 | <b>22.37</b> | 19.12 | <b>21.37</b> | 18.01 | <b>20.51</b> | 16.83 | <b>19.08</b> |
| NDCG@20   | 25.11    | <b>28.40</b> | 25.09 | <b>27.86</b> | 24.41 | <b>27.26</b> | 23.74 | <b>26.67</b> | 23.11 | <b>25.69</b> | 21.91 | <b>24.69</b> | 19.83 | <b>22.80</b> |
| NDCG@50   | 31.80    | <b>35.10</b> | 31.78 | <b>34.44</b> | 30.90 | <b>33.73</b> | 30.15 | <b>33.06</b> | 29.40 | <b>32.04</b> | 27.94 | <b>30.88</b> | 25.62 | <b>28.83</b> |
| Dateset   | Citeseer |              |       |              |       |              |       |              |       |              |       |              |       |              |
| AIR       | 10%      |              | 20%   |              | 30%   |              | 40%   |              | 50%   |              | 60%   |              | 70%   |              |
| Method    | SAT      | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         |
| Recall@10 | 7.27     | <b>9.69</b>  | 7.23  | <b>9.48</b>  | 7.24  | <b>9.09</b>  | 6.95  | <b>8.79</b>  | 6.61  | <b>8.44</b>  | 5.84  | <b>7.78</b>  | 5.68  | <b>7.19</b>  |
| Recall@20 | 12.33    | <b>15.50</b> | 11.95 | <b>15.12</b> | 11.94 | <b>14.59</b> | 11.52 | <b>14.00</b> | 11.03 | <b>13.44</b> | 9.95  | <b>12.51</b> | 9.97  | <b>11.54</b> |
| Recall@50 | 22.90    | <b>26.36</b> | 22.28 | <b>25.87</b> | 22.55 | <b>25.16</b> | 21.75 | <b>24.23</b> | 20.88 | <b>23.46</b> | 19.20 | <b>22.14</b> | 18.99 | <b>20.58</b> |
| NDCG@10   | 12.61    | <b>16.50</b> | 12.40 | <b>16.16</b> | 12.25 | <b>15.59</b> | 11.85 | <b>15.11</b> | 11.29 | <b>14.47</b> | 9.71  | <b>13.38</b> | 8.63  | <b>12.51</b> |
| NDCG@20   | 16.83    | <b>21.37</b> | 16.34 | <b>20.88</b> | 16.17 | <b>20.18</b> | 15.65 | <b>19.46</b> | 14.98 | <b>18.64</b> | 13.13 | <b>17.33</b> | 12.21 | <b>16.14</b> |
| NDCG@50   | 23.74    | <b>28.50</b> | 23.09 | <b>27.94</b> | 23.11 | <b>27.13</b> | 22.35 | <b>26.18</b> | 21.41 | <b>25.21</b> | 19.19 | <b>23.67</b> | 18.10 | <b>22.06</b> |
| Dateset   | Amac     |              |       |              |       |              |       |              |       |              |       |              |       |              |
| AIR       | 10%      |              | 20%   |              | 30%   |              | 40%   |              | 50%   |              | 60%   |              | 70%   |              |
| Method    | SAT      | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         |
| Recall@10 | 3.81     | <b>4.50</b>  | 3.81  | <b>4.47</b>  | 3.78  | <b>4.44</b>  | 3.75  | <b>4.39</b>  | 3.70  | <b>4.36</b>  | 3.59  | <b>4.26</b>  | 3.24  | <b>4.14</b>  |
| Recall@20 | 6.88     | <b>7.88</b>  | 6.82  | <b>7.83</b>  | 6.79  | <b>7.76</b>  | 6.70  | <b>7.69</b>  | 6.69  | <b>7.60</b>  | 6.52  | <b>7.47</b>  | 6.13  | <b>7.39</b>  |
| Recall@50 | 14.86    | <b>16.24</b> | 14.79 | <b>16.21</b> | 14.73 | <b>16.10</b> | 14.64 | <b>16.02</b> | 14.52 | <b>15.91</b> | 14.28 | <b>15.72</b> | 13.99 | <b>15.83</b> |
| NDCG@10   | 9.38     | <b>10.97</b> | 9.37  | <b>10.94</b> | 9.32  | <b>10.87</b> | 9.26  | <b>10.75</b> | 9.09  | <b>10.65</b> | 8.86  | <b>10.44</b> | 8.75  | <b>10.17</b> |
| NDCG@20   | 13.49    | <b>15.45</b> | 13.42 | <b>15.39</b> | 13.36 | <b>15.29</b> | 13.23 | <b>15.14</b> | 13.09 | <b>14.98</b> | 12.79 | <b>14.73</b> | 12.40 | <b>14.53</b> |
| NDCG@50   | 22.00    | <b>24.33</b> | 21.92 | <b>24.30</b> | 21.84 | <b>24.16</b> | 21.70 | <b>23.99</b> | 21.47 | <b>23.81</b> | 21.11 | <b>23.50</b> | 20.91 | <b>23.53</b> |
| Dateset   | Amap     |              |       |              |       |              |       |              |       |              |       |              |       |              |
| AIR       | 10%      |              | 20%   |              | 30%   |              | 40%   |              | 50%   |              | 60%   |              | 70%   |              |
| Method    | SAT      | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         | SAT   | RITR         |
| Recall@10 | 3.95     | <b>4.40</b>  | 3.90  | <b>4.37</b>  | 3.92  | <b>4.35</b>  | 3.90  | <b>4.30</b>  | 3.86  | <b>4.27</b>  | 3.82  | <b>4.25</b>  | 3.53  | <b>4.12</b>  |
| Recall@20 | 7.16     | <b>7.86</b>  | 7.13  | <b>7.82</b>  | 7.13  | <b>7.76</b>  | 7.05  | <b>7.71</b>  | 6.97  | <b>7.64</b>  | 6.93  | <b>7.50</b>  | 6.43  | <b>7.26</b>  |
| Recall@50 | 15.49    | <b>16.48</b> | 15.42 | <b>16.44</b> | 15.44 | <b>16.36</b> | 15.34 | <b>16.29</b> | 15.16 | <b>16.19</b> | 15.20 | <b>16.00</b> | 14.14 | <b>15.39</b> |
| NDCG@10   | 9.68     | <b>10.80</b> | 9.60  | <b>10.75</b> | 9.63  | <b>10.67</b> | 9.57  | <b>10.57</b> | 9.48  | <b>10.48</b> | 9.39  | <b>10.42</b> | 8.71  | <b>10.14</b> |
| NDCG@20   | 13.98    | <b>15.39</b> | 13.92 | <b>15.33</b> | 13.93 | <b>15.22</b> | 13.81 | <b>15.11</b> | 13.65 | <b>14.96</b> | 13.57 | <b>14.78</b> | 12.62 | <b>14.34</b> |
| NDCG@50   | 22.86    | <b>24.55</b> | 22.76 | <b>24.50</b> | 22.80 | <b>24.37</b> | 22.66 | <b>24.25</b> | 22.41 | <b>24.09</b> | 22.39 | <b>23.84</b> | 20.89 | <b>23.00</b> |

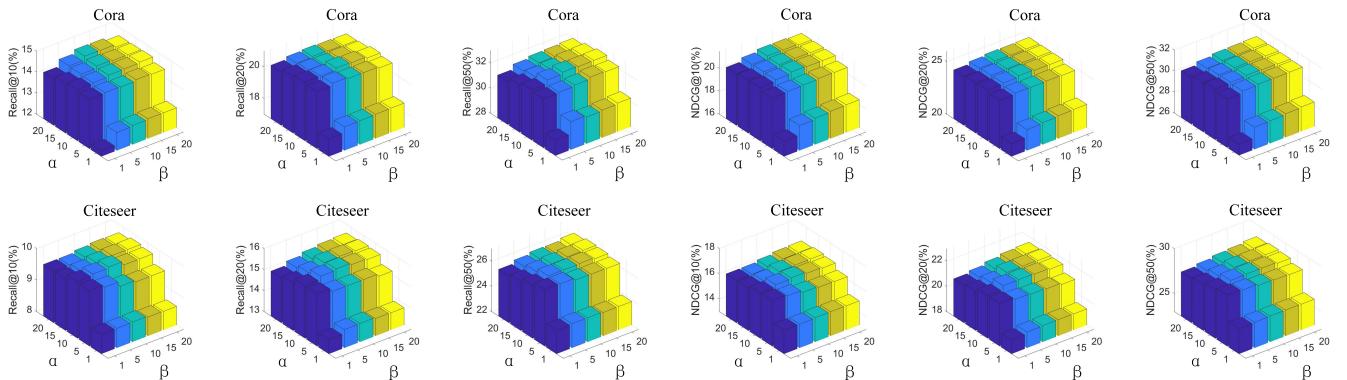


Fig. 5. The sensitivity analysis of RITR with the variation of two hyper-parameters. Both attribute-incomplete and attribute-missing ratios are set to 60%.

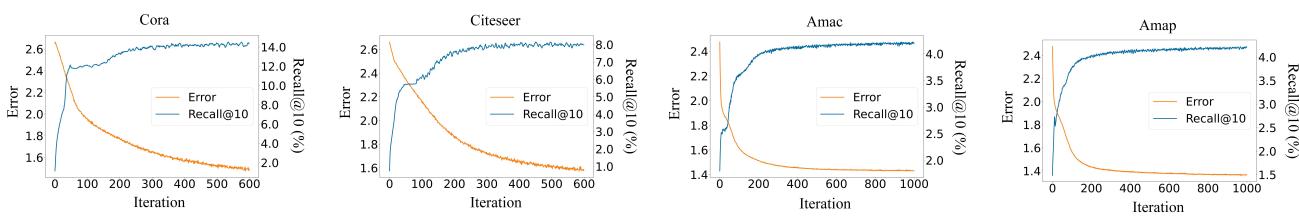


Fig. 6. Illustration of method convergence and performance variation of RITR. X-axis, left Y-axis, and right Y-axis refer to the iteration number, the final objective error, and the Recall@10 performance, respectively. Both attribute-incomplete and attribute-missing ratios are set to 60%.

As shown, the performance of the model with a certain  $\beta$  value has similar trends when we change the  $\alpha$  value; 3) RITR tends to perform well by setting  $\alpha$  and  $\beta$  to 10 according to the results of all datasets.

5) *Convergence and Performance Variation (Q5):* To illustrate the convergence of the proposed RITR, we record the profiling performance reflected by the Recall@10 metric and plot the objective error of RITR with iterations on four datasets. From these sub-figures illustrated in Fig. 6, we can observe that 1) the Recall@10 metric of RITR first gradually increases to a plateau with an obvious tendency and then keeps stable with a wide range of iterations; 2) RITR can converge within 1000 epochs on four datasets. These results clearly verify the good convergence property of our proposed method and reveal the effectiveness of the learning procedure.

## V. CONCLUSION AND FUTURE WORK

Hybrid-absent graphs are ubiquitous in practical applications. However, the corresponding learning problem that significantly influences the performance of existing graph machine learning methods is still left under-explored. We firstly propose ITR toward the attribute-missing circumstance, which can enable the attribute and structure information to sufficiently negotiate with each other for accurate missing value restoration. We further improve ITR and design a variant called RITR to handle hybrid-absent graphs, which can effectively leverage the intimate structure-attribute relationship to guide the imputation of incomplete attributes and employ the most trustworthy visible information to implement the missing attribute completion. Extensive experiments on four benchmark datasets have been conducted to compare two proposed methods with state-of-the-art competitors. These results have solidly demonstrated the superiority and robustness of ITR and RITR on both profiling and node classification tasks. However, some limitations of existing attribute-missing or hybrid-absent graph machine learning methods are we still have not explored some issues for them. For instance, the time complexities of most methods are  $\mathcal{O}(N^2)$ , making them hard to be deployed to various large-scale graph-oriented applications. Future work may extend the proposed RITR to a scalable version with linear scalability (*i.e.*,  $\mathcal{O}(BNd)$ ) via a mini-batch design. Moreover, in the current version, RITR conducts the structure-attribute information interaction and imputation via a simple concatenation. In the future, how to develop a more mathematical hybrid-absent graph machine learning approach to illustrate the structure-attribute relationship in theory and alleviate the mutual interference between them is another interesting direction, which may further improve the quality of data imputation and graph representations.

## REFERENCES

- [1] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, “Hierarchical representation learning in graph neural networks with node decimation pooling,” *IEEE TNNLS*, vol. 33, no. 5, pp. 2195–2207, 2022.
- [2] J. Chen, Z. Gong, W. Wang, C. Wang, Z. Xu, J. Lv, X. Li, K. Wu, and W. Liu, “Adversarial caching training: Unsupervised inductive network representation learning on large-scale graphs,” *IEEE TNNLS*, vol. 33, no. 12, pp. 7079–7090, 2022.
- [3] H. Huang, Y. Song, Y. Wu, J. Shi, X. Xie, and H. Jin, “Multitask representation learning with multiview graph convolutional networks,” *IEEE TNNLS*, vol. 33, no. 3, pp. 983–995, 2022.
- [4] J. Liu, F. Xia, X. Feng, J. Ren, and H. Liu, “Deep graph learning for anomalous citation detection,” *IEEE TNNLS*, vol. 33, no. 6, pp. 2543–2557, 2022.
- [5] C. Chen, K. Li, W. Wei, J. T. Zhou, and Z. Zeng, “Hierarchical graph neural networks for few-shot learning,” *IEEE TNNLS*, vol. 32, no. 1, pp. 240–252, 2022.
- [6] R. Zhang, Y. Zhang, and X. Li, “Unsupervised feature selection via adaptive graph learning and constrain,” *IEEE TNNLS*, vol. 33, no. 3, pp. 1355–1362, 2022.
- [7] Z. Li, H. Liu, Z. Zhang, T. Liu, and N. N. Xiong, “Learning knowledge graph embedding with heterogeneous relation attention,” *IEEE TNNLS*, vol. 33, no. 8, pp. 3961–3973, 2022.
- [8] R. van den Berg, T. N. Kipf, and M. Welling, “Graph convolutional matrix completion,” in *ArXiv abs/1706.02263*, 2017.
- [9] M. Zhang and Y. Chen, “Inductive matrix completion based on graph neural networks,” in *ICLR*, 2020.
- [10] I. Spinelli, S. Scardapane, and A. Uncini, “Missing data imputation with adversarially-trained graph convolutional networks,” *NN*, vol. 129, pp. 249–260, 2020.
- [11] H. Taguchi, X. Liu, and T. Murata, “Graph convolutional networks for graphs containing missing features,” *FGCS*, vol. 117, pp. 155–168, 2021.
- [12] W. Shen, C. Zhang, Y. Tian, L. Zeng, X. He, W. Dou, and X. Xu, “Inductive matrix completion using graph autoencoder,” in *CIKM*, 2021, pp. 1609–1618.
- [13] X. Xie, N. Zhang, Z. Li, S. Deng, H. Chen, F. Xiong, M. Chen, and H. Chen, “From discrimination to generation: Knowledge graph completion with generative transformer,” in *WWW*, 2022.
- [14] X. Chen, S. Chen, J. Yao, H. Zheng, Y. Zhang, and I. W. Tsang, “Learning on attribute-missing graphs,” *TPAMI*, vol. 44, no. 2, pp. 740–757, 2022.
- [15] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *ICLR*, 2017.
- [16] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *ICLR*, 2018.
- [17] W. Tu, S. Zhou, X. Liu, Y. Liu, Z. Cai, E. Zhu, C. Zhang, and J. Cheng, “Initializing then refining: A simple graph attribute imputation network,” in *IJCAI*, 2022.
- [18] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *SIGKDD*, 2014, pp. 701–710.
- [19] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *SIGKDD*, 2016, pp. 855–864.
- [20] J. Song, X. Luo, J. Gao, C. Zhou, H. Wei, and J. X. Yu, “Uniwalk: Unidirectional random walk based scalable simrank computation over large graph,” *IEEE TKDE*, vol. 30, no. 5, pp. 992–1006, 2018.
- [21] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” in *ArXiv abs/1611.07308*, 2016.
- [22] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, “Structural deep clustering network,” in *WWW*, 2020, pp. 1400–1410.
- [23] W. Tu, S. Zhou, X. Liu, X. Guo, Z. Cai, E. Zhu, and J. Cheng, “Deep fusion clustering network,” in *AAAI*, 2021, pp. 9978–9987.
- [24] Y. Liu, W. Tu, S. Zhou, X. Liu, L. Song, X. Yang, and E. Zhu, “Deep graph clustering via dual correlation reduction,” in *AAAI*, 2022.
- [25] Z. Tao, H. Liu, J. Li, Z. Wang, and Y. Fu, “Adversarial graph embedding for ensemble clustering,” in *IJCAI*, 2019, pp. 3562–3568.
- [26] F. Feng, X. He, J. Tang, and T.-S. Chua, “Graph adversarial training: Dynamically regularizing based on graph structure,” *IEEE TKDE*, vol. 33, no. 6, pp. 2493–2504, 2021.
- [27] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, W. Li, X. Xie, and M. Guo, “Learning graph representation with generative adversarial nets,” *IEEE TKDE*, vol. 33, no. 8, pp. 3090–3103, 2021.
- [28] S. Pan, R. Hu, S.-F. Fung, G. Long, J. Jiang, and C. Zhang, “Learning graph embedding with adversarial training methods,” *IEEE TCYB*, vol. 50, no. 6, pp. 2475–2487, 2020.
- [29] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, “Deep graph infomax,” in *ICLR*, 2019.
- [30] K. Hassani and A. H. K. Ahmadi, “Contrastive multi-view representation learning on graphs,” in *ICML*, 2020, pp. 4116–4126.
- [31] Z. Peng, W. Huang, M. Luo, Q. Zheng, Y. Rong, T. Xu, and J. Huang, “Graph representation learning via graphical mutual information maximization,” in *WWW*, 2020, pp. 259–270.
- [32] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, “Graph contrastive learning with adaptive augmentation,” in *WWW*, 2021, pp. 2069–2080.
- [33] L. Gong, S. Zhou, W. Tu, and X. Liu, “Attributed graph clustering with dual redundancy reduction,” in *IJCAI*, 2022.

- [34] Z. Zhou, Y. Hu, Y. Zhang, J. Chen, and H. Cai, “Multiview deep graph infomax to achieve unsupervised graph embedding,” *IEEE TCYB*, 2022.
- [35] G. Ian, P.-A. Jean, M. Mehdi, X. Bing, W.-F. David, O. Sherjil, C. Aaron, and B. Yoshua, “Generative adversarial nets,” in *NeurIPS*, 2014.
- [36] C. M. Bishop, *Pattern Recognition and Machine Learning*. Stat Sci, 2006.
- [37] C. Gao, X. He, D. Gan, X. Chen, F. Feng, Y. Li, T. Chua, and D. Jin, “Neural multi-task recommendation from multi-behavior data,” in *ICDE*, 2019, p. 1554–1557.
- [38] J. You, X. Ma, D. Y. Ding, M. J. Kochenderfer, and J. Leskovec, “Handling missing data with graph representation learning,” in *NeurIPS*, 2020.
- [39] 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,” *arXiv abs/2212.12738*, 2022.
- [40] D. Jin, C. Huo, C. Liang, and L. Yang, “Heterogeneous graph neural network via attribute completion,” in *WWW*, 2021, pp. 391–400.
- [41] D. He, C. Liang, C. Huo, Z. Feng, D. Jin, L. Yang, and W. Zhang, “Analyzing heterogeneous networks with missing attributes by unsupervised contrastive learning,” *IEEE TNNLS*, 2022.
- [42] J. Yoo, H. Jeon, J. Jung, and U. Kang, “Accurate node feature estimation with structured variational graph autoencoder,” in *Proceedings of the Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2336–2346.
- [43] D. Jin, R. Wang, T. Wang, D. He, W. Ding, Y. Huang, L. Wang, and W. Pedrycz, “Amer: A new attribute-missing network embedding approach,” *IEEE TCYB*, 2022.
- [44] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *ICML*, 2008, pp. 1096–1103.
- [45] H. Gao and H. Huang, “Deep attributed network embedding,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 3364–3370.
- [46] Özgür Simsek and D. D. Jensen, “Navigating networks by using homophily and degree,” *NAS*, vol. 105, no. 35, pp. 12 758–12 762, 2008.
- [47] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *ICLR*, 2014.
- [48] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *ICLR*, 2017.
- [49] W. L. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *NeurIPS*, 2017, pp. 1024–1034.
- [50] X. Huang, Q. Song, Y. Li, and X. Hu, “Graph recurrent networks with attributed random walks,” in *SIGKDD*, 2019, pp. 732–740.
- [51] L. Chen, S. Gong, J. Bruna, and M. M. Bronstein, “Attributed random walk as matrix factorization,” in *NeurIPS Workshop*, 2019.
- [52] 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 *AAAI*, 2019, pp. 3830–3837.