Robust Attribute and Structure Preserving Graph Embedding

Anonymous Authors

No Institute Given

Abstract. Graph embedding methods are useful for a wide range of graph analysis tasks including link prediction and node classification. Most graph embedding methods learn only topological structure of graphs. Nevertheless, it has been shown that the incorporation of node attributes is beneficial in improving the expressive power of node embeddings. However, real-world graphs are often noisy in terms of structure and/or attributes (missing and/or erroneous edges/attributes). Most existing graph embedding methods are susceptible to this noise, as they do not consider uncertainty during the modelling process. In this paper, we introduce RASE, a **R**obust **A**ttribute and **S**tructure preserving graph **E**mbedding model. RASE is a novel graph representation learning model which effectively preserves both graph structure and node attributes through a unified loss function. To be robust, RASE uses a denoising attribute auto-encoder to deal with node attribute noise, and models uncertainty in the embedding space as Gaussian embeddings to cope with graph structure noise. We evaluate the performance of RASE through an extensive experimental study on various real-world datasets. Results demonstrate that RASE outperforms state-of-the-art embedding methods on multiple graph analysis tasks, and is robust to both structure and attribute noise.

Keywords: Robust Graph Embedding·Node Classification·Link Prediction.

1 Introduction

Much real-world data can be naturally delineated as graphs, e.g. citation networks [1,5,13], social-media datasets [6] and language networks [13]. Graph embedding methods [4,5,10,13] have been proposed as an effective way of learning low-dimensional representations for nodes to enable down-stream machine learning tasks, such as link prediction, node classification and visualization, on these complex graph data. Most existing graph embedding methods learn low-dimensional representations from graph topological structure only [4,10,13]. However, nodes in a graph usually have rich information, and these supplementary attributes can be utilized in graph embedding along with the graph structure to produce more meaningful node embeddings [1,2,5,8,12,18].

Graphs constructed from the real-world data are usually non-deterministic and ambiguous [11], manifested by uncertain and ambiguous edges/node-attributes. Thus, the real-world graphs may be corrupted with missing/erroneous edges and missing/erroneous node attributes. For example, most knowledge graphs, originating from the Semantic Web, follow the "Open World Assumption" [11] (i.e. the edges unobserved are simply unknown instead of untrue), which means that graph structures

are far from complete and many edges are missing. Also, much attribute information is abstracted from free text (e.g. users' post on social media, text information on Semantic Web etc.) and is usually imprecise or ambiguous due to the limitation of data sources or abstraction tools. We term this non-deterministic and ambiguous phenomena in graph structure and node attributes as **"structure noise"** and **"attribute noise"** respectively.

A great challenge the existing graph embedding methods face when incorporating both graph structure and node attributes in embedding learning, is the noise prevalent in these two aspects which can mislead the embedding technique to result in learning invalid latent information. Recently, several work has been proposed to model the uncertainty present in graph data [1,16,17]. Most of these work, including VGAE [8], Graph2Gauss [1] and DVNE [17], focus on modelling the uncertainty of the node embeddings by representing the nodes with a probabilistic distribution in the embedding space. Since these studies focus on preserving graph structural proximity by measuring the distance between probability distribution embeddings, uncertainty modelling of these methods can only capture structure noise. Therefore, they do not explicitly account for the attribute noise which is common in the real-world graphs.

In this work, we introduce RASE, a novel graph embedding framework to address the aforementioned challenges. RASE learns robust node representations, exploiting both graph structure and node attributes simultaneously. To be robust to noisy real-world graphs, carefully-designed strategies have been introduced to model both structure and attribute noise in RASE. Attribute noise is modelled with a denoising attribute auto-encoder to maintain the discreteness and sparseness of textual data via introducing noise in the input through a binomial distribution. On the other hand, structure noise is modelled in the latent layer via modelling the latent representations with a Gaussian distribution. To preserve the transitivity in the embedding space with a linear computational cost, 2-Wasserstein distance is used as the similarity measure between the distributions in Gaussian space. Extensive experiments have been conducted on five different real-world datasets. The experimental results show that our method significantly outperforms the state-of-art methods in generating effective embeddings for node classification, link prediction and visualisation. Moreover, we introduce a novel experimental setting which simulates random structure noise and random attribute noise to demonstrate the robustness of our model in embedding noisy graphs.

2 Related Work

There are three lines of effort most related to this work: structure preserving graph embedding, attributed graph embedding and noise modelled graph embedding.

Structure preserving graph embedding: These embedding methods attempt to conserve observable graph structure properties in the embedding space. LINE [13] learns from structural closeness considering first- (connected nodes) and second-order (neighbourhood nodes) proximity. DeepWalk [10] and node2vec [4] learn node embeddings from random walk sequences with a technique similar to Skip-Gram [9]. DVNE [17] uses an auto-encoder architecture to encode and reconstruct the graph structure. All these algorithms focus on graph structure only.

Attributed graph embedding: Recent studies [1,5,6,8,12,15,16,18] show that the incorporation of node attributes along with graph structure produces better node embeddings. TADW [15] incorporates text attributes and graph structure with low-rank matrix factorization. GraphSAGE [5] is a CNN-based technique that samples and aggregates neighbouring node attributes. Graph2Gauss [1] captures multi-hop neighbours and uses attributes for embedding initialization. VGAE [8] is a graph convolution network (GCN) method, which aggregates neighbouring attributes. In most studies, node attributes are only used for node embedding initialization, but not during model training. DANE [2] proposes a deep non-linear architecture to preserve both aspects.

Noise modelled graph embedding: Attributed graphs can be noisy in terms of graph structure and node attributes. Most of the existing graph embedding methods represent nodes as point vectors in the embedding space, ignoring the uncertainty of the embeddings. In contrast, Graph2Gauss [1], VGAE [8], and DVNE [17] capture the uncertainty of graph structure by learning node embeddings as Gaussian distributions. DVNE [17] proposes to measure distributional distance using the Wasserstein metric as it preserves transitivity, while the other two uses KL divergence. However, these works ignore the modelling of uncertainty of node attributes. SINE [16] is a recent work on incomplete graphs which focuses only on missing structure and discrete node attributes.

3 Methodology

Problem Formulation Let $G=(V,E,X)$ be an **attributed graph**, where V is the set of nodes, E is the set of edges in which each ordered pair of nodes $(i,j) \in E$ is associated with a weight $w_{ij} > 0$ for edge from i to j, and $X_{|V| \times D}$ is the node attribute matrix, where $\mathbf{x}_i \in X$ is a D-dimensional attribute vector of node i. We learn to embed each node $i \in V$ as a low-dimensional Gaussian distribution $\mathbf{z}_i = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$[1], where $\boldsymbol{\mu}_i \in \mathbb{R}^L$, $\boldsymbol{\sigma}_i^2 \in \mathbb{R}^{L \times L}$ with the embedding dimension $L \ll |V|$, D. The learning goal is such that, nodes that are closer in the graph and have similar attributes, are closer in the embedding space, and node embeddings are robust to structure noise and attribute noise.

3.1 RASE Architecture

Figure 1 shows the architecture of RASE which is an end-to-end embedding framework that learns from both node attributes and graph structure, with two main components: *Node Attribute Learning* and *Graph Structure Learning*. To deal with attribute noise, RASE corrupts node attributes by introducing a random noise ε_i sampled from a binomial distribution, which are then projected to a low-dimensional intermediate representation $\mathbf{u}_i$. RASE takes this $\mathbf{u}_i$ as input and simultaneously performs node attribute learning and graph structure learning. By reconstructing the node attributes from $\mathbf{u}_i$, the model preserves attributes (with Euclidean distance to preserve transitivity) while being robust to attribute noise. RASE models uncertainty of the graph structure noise by learning Gaussian embeddings and capturing neighbourhood information measured with Wasserstein metric to preserve transitivity property in the embedding space.

[1] We learn $\boldsymbol{\sigma}_i^2$ as a diagonal covariance vector, $\boldsymbol{\sigma}_i \in \mathbb{R}^L$, instead of a covariance matrix to reduce the number of parameters to learn.

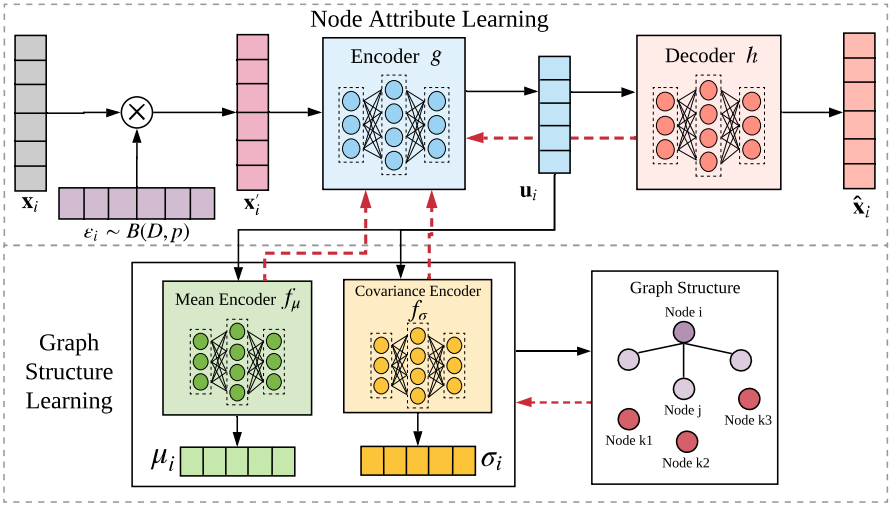


Fig. 1: RASE architecture.

Node Attribute Learning We learn node attributes with an unsupervised learning function (as opposed to semi-supervised GCN methods). To deal with missing and noisy attributes, we slightly corrupt the attribute vectors using random noise. In most real-world graphs, node attributes can be very sparse, since they are either tf-idf vectors of text features or one-hot vectors of categorical features. Therefore, we pull noise from a binomial distribution, to represent a masking noise but still depict the trends in the original data. Accordingly, we inject some impurity to the original node attribute vector $\mathbf{x}_i \in \mathbb{R}^D$ by sampling a random binary noise vector $\varepsilon_i \in \{0,1\}^D$ from a binomial distribution B with D (i.e. attribute vector dimension) trials and p success probability. We set $p \in (0.90, 0.98)$ to ensure that the noise is small and its introduction does not change original data trends. We produce the corrupted attribute vector $\mathbf{x}'_i \in \mathbb{R}^D$ by performing Hadamard product: $\mathbf{x}'_i = \mathbf{x}_i \otimes \varepsilon_i$.

The corrupted attribute vector is transformed into an intermediate representation $\mathbf{u}_i \in \mathbb{R}^m$ where m is a reduced vector dimension using an encoding transformation function, $g : \mathbb{R}^D \to \mathbb{R}^m$. Subsequently, this intermediate vector is fed as input to a decoder, $h : \mathbb{R}^m \to \mathbb{R}^D$, to reconstruct the attribute vector $\hat{\mathbf{x}}_i \in \mathbb{R}^D$. Note here that, these encoder and decoder functions can easily be implemented with MLP layers or sophisticated GCN layers [8], and to capture the non-linearity in data we can have deep neural networks. But we observe that MLP architecture is more simple and efficient, hence scalable on large-scale graphs. We define the attribute reconstruction loss as the Euclidean distance between the original and reconstructed attributes:

$$\hat{\mathbf{x}}_i = h(\mathbf{u}_i) = h(g(\mathbf{x}'_i)) \tag{1}$$

$$\mathcal{L}_a = \sum_{i \in V} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 \tag{2}$$

L1 regularization has been adopted as we have sparse attribute vectors constructed from textual data. By minimizing the attribute reconstruction loss we encourage the *encoder* g to generate robust latent representations, $\mathbf{u}_i$, preserving attribute information which are used as inputs to the *Graph Structure Learning* component.

Graph Structure Learning We use the intermediate vector, $\mathbf{u}_i$, from the auto-encoder in the *Node Attribute Learning* component, as it encodes attribute latent relationships between nodes. We define two parallel transformation functions to model a node's embedding as a Gaussian distribution representation to account for structural uncertainty (due to noise). Functions f_μ and f_σ learn the mean vector $\boldsymbol{\mu}_i$ and the diagonal covariance vector $\boldsymbol{\sigma}_i$ of $\mathbf{u}_i$ respectively. Again, we have the flexibility to select either MLP or CNN based architecture for these functions. To obtain positive $\boldsymbol{\sigma}_i$ for interpretable uncertainty we choose activation function at the output layer accordingly. Thus, the final latent representation of node i is $\mathbf{z}_i = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$, where:

$$\boldsymbol{\mu}_i = f_\mu(\mathbf{u}_i) \text{ and } \boldsymbol{\sigma}_i = f_\Sigma(\mathbf{u}_i) \tag{3}$$

To preserve the structural proximity of nodes in the graph, we assume that the nodes which are connected with a higher edge weight are more likely to be similar. Therefore, we attempt to pull the embeddings of these nodes closer in the embedding space. We define the prior probability for connected nodes as $\hat{P}(i,j) = \frac{w_{ij}}{\Sigma_{(\hat{i},\hat{j})\in E} w_{\hat{i}\hat{j}}}$. Since RASE's node embeddings are Gaussians, we choose a probability distance metric to compute the distance between nodes. Thus, motivated by DVNE [17], to preserve transitivity property in the embedding space, we choose the Wasserstein distance: 2-nd moment (W_2). This metric allows to discover specific relations between nodes based on their semantic relations and similarities by leveraging the geometric properties of the embedding space. As a result, when we model the explicit local neighbourhood edges, implicit global neighbourhood proximity can be modelled due to triangle inequality property. We define $\delta(\mathbf{z}_i, \mathbf{z}_j)$ as the W_2 distance for our embeddings of nodes, i and j. Modelling only the diagonal covariance vectors results in $\boldsymbol{\sigma}_i \boldsymbol{\sigma}_j = \boldsymbol{\sigma}_j \boldsymbol{\sigma}_i$. Hence, the distance computation [3] simplifies to:

$$\delta(\mathbf{z}_i, \mathbf{z}_j) = W_2(\mathbf{z}_i, \mathbf{z}_j) = (\| \boldsymbol{\mu}_i - \boldsymbol{\mu}_j \|_2^2 + \| \boldsymbol{\sigma}_i - \boldsymbol{\sigma}_j \|_F^2)^{1/2} \tag{4}$$

The observed joint probability between nodes i and j is defined as:

$$P(i,j) = Sigmoid(-\delta(\mathbf{z}_i, \mathbf{z}_j))) = \frac{1}{1+\exp(\delta(\mathbf{z}_i, \mathbf{z}_j))} \tag{5}$$

We minimize the distance between the prior and observed probability distributions for all edges observed in the graph. Since $\hat{P}$ and P are discrete distributions, we define structural loss function:

$$\mathcal{L}_s = D_{KL}(\hat{P}||P) = \sum_{(i,j)\in E} \hat{P}(i,j) \log\Big(\frac{\hat{P}(i,j)}{P(i,j)}\Big) \propto -\sum_{(i,j)\in E} \hat{P}(i,j) \log P(i,j) \propto -\sum_{(i,j)\in E} w_{ij} \log P(i,j) \tag{6}$$

For regularization of $\mathcal{L}_s$, instead of regularizing mean and covariance functions separately [14], RASE uses the strategy similar to [7] minimising KL divergence between the learned Gaussian representation and the standard normal distribution. Thus, it will ensure that the final latent space will be closer to a standard Gaussian space other than pushing values in both mean vectors and variance vectors to be small. The regularization for node i is:

$$D_{KL}(\mathbf{z}_i || \mathcal{N}(\mathbf{0}, \mathbf{1})) = \frac{1}{2}\Big(tr(\boldsymbol{\sigma}_i) + \boldsymbol{\mu}_i^T \cdot \boldsymbol{\mu}_i - L - \log(det(\boldsymbol{\sigma}_i))\Big) \tag{7}$$

By minimizing the overall structural loss function, we attempt to construct an embedding space where nodes that are similar in terms of graph structure are also similar in the embedding space, and robust to noisy graph structure.

Unified Training and Optimization To jointly preserve node attributes and graph structure, we define a unified loss function by combining Eq. 2 and Eq. 6 with hyperparameter α. For simplicity, we omit the regularization terms in the two components of RASE in the overall loss function to be minimized:

$$\mathcal{L} = \alpha\mathcal{L}_a + \mathcal{L}_s \quad = \alpha \sum_{i \in V} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 - \sum_{(i,j) \in E} w_{ij} \log P(i,j) \tag{8}$$

For large graphs, this unified loss function is computationally expensive, since it has to compute the attribute reconstruction loss ($\mathcal{L}_a$) for all the nodes, and the structural loss ($\mathcal{L}_s$) for all the edges. To optimize $\mathcal{L}_a$, we sample only a batch of nodes in each epoch. To optimize $\mathcal{L}_s$, we employ the negative sampling approach proposed in Skip-Gram [9] and sample K negative edges for each edge in the training batch.

4 Experiments

We evaluate RASE against state-of-the-art baselines in several graph analysis tasks, including node classification, link prediction and robustness on several public datasets. Two additional tasks, visualisation and uncertainty modelling, can be found in the supplementary materials. Source code of RASE and the datasets will be made available upon publication.

4.1 Datasets

Social Media Networks [6]: BlogCatalog and Flickr (cf Table 1). Nodes on these social media networks are users. The following relationships are used to construct the edges. Attributes on BlogCatalog and Flickr are constructed with keywords in users' blog description and users' predefined tags of interests, respectively. Node labels are users' interest topics on BlogCatalog and groups users joined in Flickr.

Citation Networks [1]: Cora , Citeseer and Pubmed (cf Table 1). Nodes denote papers and edges represent citation relations. We use tf-idf word vectors of the paper's abstract as node attributes. Each paper is assigned a label based on the topic of the paper.

4.2 Compared Algorithms

We compare RASE to several state-of-the-art graph embedding methods: structure-based non-attributed embedding methods (node2vec, LINE and DVNE); attributed embedding methods (GraphSAGE, VGAE and Graph2Gauss); and uncertainty modelling embedding methods (DVNE, VGAE and Graph2Gauss).

node2vec [4] is a random walk based node embedding method that maximizes the likelihood of preserving nodes' neighbourhood using biased random walks. **LINE** [13]

Table 1: Statistics of the real-world graphs.

Dataset	$\|V\|$	$\|E\|$	D	#Labels
Social media networks				
BlogCatalog	5,196	369,435	8,189	6
Flickr	7,535	239,738	12,047	9
Citation networks				
Cora	2,995	8,416	2,879	7
Citeseer	4,230	5,358	602	6
Pubmed	18,230	79,612	500	3

preserves first-order and second-order proximity information. We report results on a concatenated representation of the two proximities (as suggested). **DVNE** [17] learns a Gaussian distribution in the Wasserstein space for plain graphs. **GraphSAGE** [5] is an attributed embedding method for graphs which learns by sampling and aggregating features of local neighbourhoods. We use the unsupervised version of GraphSAGE (since all other methods are unsupervised) with the pooling aggregator which performs best for citation networks in the paper. **VGAE** [8] is an attributed GCN-based embedding method which implements an auto-encoder model with Gaussian node embeddings. **Graph2Gauss (G2G)** [1] is an attributed embedding method which represents each node as a Gaussian distribution and preserves graph structure using a personalized ranking of nodes based on multiple neighbouring hops. In addition, we also evaluate task performance on **Attributes**, raw node attributes as input features instead of learning node embeddings, for down-stream machine learning tasks.

RASE is our full model which jointly preserves node attribute and graph structure, and is robust to noise in real-world graphs. We also consider a non-robust version, **RASE($\neg\boldsymbol{R}$)**, for an ablation study. **RASE($\neg\boldsymbol{R}$)** does not model attribute noise and learns point vectors, thus also ignoring structural uncertainty.

4.3 Experimental Settings

For all the models that learn point vectors, we set $L=128$ as the embedding dimension. For a fair evaluation, we set $L=64$ in methods learning probability distributions as embeddings including ours, so that the parameters learned per node is still 128 ($\mu_i \in \mathbb{R}^{64}$ and $\Sigma_i \in \mathbb{R}^{64}$). The other parameters for baselines are referred from the papers and tuned to be optimal. For our two models and LINE, we set the number of negative samples as $K=5$. The hyperparameter α is tuned to be optimal using grid search on a validation set. We used the Adam optimizer with a learning rate 0.001. We report the results averaged over 10 trials.

4.4 Node Classification

In this task, each method learns the embeddings in an unsupervised manner, and a logistic regression (LR) classifier is trained on these embeddings to classify each node into their associated class label. We randomly sample different percentages of labeled

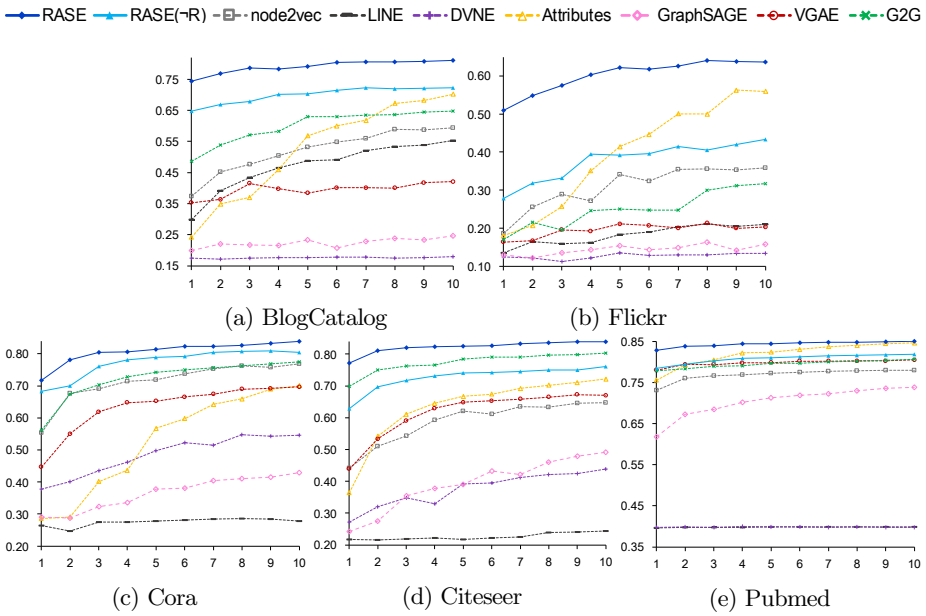


Fig. 2: Node classification performance measured by micro-F1 score (y-axis) in terms of percentage of labelled nodes (x-axis). RASE's improvements are statistically significant for $p<0.01$ by a paired t-test.

nodes (i.e. 1%,2%,...,10%) from the graph as training set for the classifier, and use the rest for evaluation. We report micro- and macro-F1 scores which have been widely used in the evaluation of multi-class classification [13,17]. Due to brevity reasons, we present only micro-F1 scores in Figure 2, and a similar trend is observed in macro-F1 scores. We show our methods with solid lines, and baselines with dashed lines.

Based on the results in Fig. 2, we can see that RASE consistently outperforms all the baselines in all the datasets with all the training ratios. Furthermore, in all five datasets, RASE has demonstrated a larger improvement margin to the baselines when only smaller numbers of nodes are used for training, e.g., a 174.9% improvement over best performing baseline in Flickr at 1% labeled nodes. This performance improvement is due to the attribute preserving component, which learns meaningful latent representations from node attributes. Moreover, denoising the attributes in this process also helps our model to deal with scarce data which is common in the real-world graphs. Also, our proposed structure learning method has captured useful local and global node similarities (due to the transitivity-preserving property in W_2 metric).

Overall, RASE($\neg R$), which is non-probabilistic and models only point vectors, manifests superiority among the non-probabilistic methods (i.e. node2vec, LINE and GraphSAGE), consistently outperforming them in all datasets. Interestingly, on BlogCatalog, Flickr and Cora, RASE($\neg R$) also substantially outperforms the probabilistic models DVNE, VGAE and G2G. This emphasizes the effectiveness of our attribute preservation and structure learning method, even in the absence of uncertainty modelling.

Table 2: Link prediction performance.

	Cora		Citeseer		Pubmed	
Algorithm	**AUC**	**AP**	**AUC**	**AP**	**AUC**	**AP**
node2vec	79.11	77.99	79.91	82.08	91.18	91.49
LINE	79.12	78.91	71.20	72.11	75.32	76.81
DVNE	65.73	70.33	68.16	73.42	50.66	50.78
Attributes	88.06	83.66	81.53	75.60	82.98	77.71
GraphSAGE	81.76	83.19	83.33	85.38	89.43	90.90
VGAE	93.53	95.33	95.46	96.47	96.11	96.09
G2G	95.92	95.82	96.28	96.54	95.75	95.65
RASE($\neg R$)	95.42	96.18	95.60	96.25	94.54	93.84
RASE	**96.88**	**96.82**	**97.82**	**97.69**	**96.40**	**96.21**

4.5 Link Prediction

This task aims to predict future links using the current graph structure and attributes. We randomly select 20% edges and an equal number of non-edges, and combine the two as the test set. The remaining 80% are used for embedding training. Then the node embeddings are used to compute the similarity between each test node pair, to see the likelihood of a link's existence between the two nodes. In Gaussian embedding methods, we use negative Wasserstein distance (RASE, DVNE) and negative KL divergence (G2G) to rank the embedding pairs [17,1]. For other methods (RASE($\neg R$), node2vec, LINE and GraphSAGE), we use dot product similarity of node embedding for ranking, and measure AUC and AP scores [1,8]. For brevity reasons, we only present citation networks in Table 2, and the trend is similar in the social media networks.

RASE clearly outperforms the state-of-the-art embedding methods by a significant margin in all the graphs, demonstrating the effectiveness of our model in capturing structural information and attribute information via the proposed method. RASE outperforms RASE($\neg R$), showing that accounting for attribute noise and structure noise collectively is beneficial. This is also validated by the performance gain of the uncertainty modelling methods, VGAE and G2G, over this non-robust RASE($\neg R$).

Moreover, the methods that learn from graph structure only (i.e. node2vec, LINE and DVNE) are significantly outperformed by the attributed embedding methods (i.e. RASE, RASE($\neg R$), GraphSAGE, VGAE and G2G).

RASE($\neg R$) is the best performing model among the non-probabilistic methods (i.e. node2vec, LINE and GraphSAGE), demonstrating that its vectors have learnt meaningful structural similarities between nodes along with node attributes.

4.6 Robustness

We evaluate RASE and the state-of-art baselines to see how they can deal with noise in graphs. In this section, we introduce a novel evaluation task to assess the robustness of graph embeddings to *random structure and attribute noise*. We inject some noise into the graphs by intentionally corrupting the graph structure and node

Table 3: Link prediction performance in Citeseer with structural noise.

% of noisy	0%		10%		20%		30%		40%		50%	
edges added	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
node2vec	72.99	76.76	66.74	70.64	62.78	66.27	57.57	61.31	56.74	60.15	56.21	59.05
LINE	53.09	49.95	52.24	49.23	52.23	48.82	52.58	49.54	51.13	48.36	51.18	48.62
DVNE	57.89	59.43	55.99	57.06	52.89	55.19	55.47	56.61	53.15	55.36	53.58	55.39
GraphSAGE	75.48	78.16	73.86	76.50	73.09	75.36	72.93	74.87	71.81	73.39	71.29	72.12
VGAE	90.20	92.48	88.60	91.14	86.92	89.84	85.79	88.94	85.95	88.74	84.32	87.46
G2G	91.33	91.91	84.83	87.16	79.73	83.18	77.59	81.81	76.38	80.16	78.24	81.43
RASE($\neg R$)	90.10	91.24	90.18	90.58	89.72	89.26	88.81	89.40	86.99	87.71	86.10	86.46
RASE	**95.95**	**95.93**	**94.90**	**94.88**	**94.33**	**94.47**	**93.52**	**93.48**	**92.62**	**93.17**	**92.22**	**92.48**

attributes. This experiment is conducted on all datasets, and we report the results on Citeseer, since all the datasets demonstrate similar trends.

Structural noise: We corrupt the graph structure by hiding randomly selected edges 50% (to mimic *missing edges* which we also use as the test set) and randomly adding some non-existing edges (edges not in the original graph to mimic *erroneous edges*). We vary the percentage of noisy edges added to the graph from 0%-50%, and observe AUC and AP decline with the increasing noise in link prediction task. The results are presented in Table 3.

From Table 3, we can see that RASE performs the best in all the structural noise percentages, showing that it is robust to real-world noisy graph structures. In addition to this, with the increase in noise ratio from 0% to 50%, RASE's AUC degradation is only 3.8%. Also, RASE outperforms its non-robust version, RASE($\neg R$), which shows that the proposed uncertainty modelling technique to mitigate structure noise is effective. In contrast, though DVNE, VGAE and G2G also model uncertainty in the embeddings, their performance degradation is quite significant (7.4%, 6.5% and 14.3% in AUC respectively) when the noise ratio is increased from 0% to 50%. VGAE is based on GCN, which aggregates the neighbouring attributes into a convex embedding. Thus, it is heavily affected by noisy neighbours, as errors get further exaggerated. The hop-based structural ranking in G2G is sensitive to false neighbourhood formation. Furthermore, the square-exponential loss function used for pair-wise ranking in both G2G and DVNE does not have a fixed margin and pushes the distance of the negative edges to infinity with an exponentially decreasing force [1]. Hence, these methods are highly sensitive to erroneous and missing edges in the graph. In contrast, RASE is mildly affected due to its carefully designed sigmoid structural loss function and the extra information learned about the global neighbourhood via the transitivity property of W_2 metric.

Attribute noise: To evaluate the robustness of the methods to random attribute noise, we corrupt the node attribute vectors randomly. Then, we assess node classification performance of the learned embeddings on these corrupted graphs. Specifically, we sample a masking noise from a binomial distribution with D (i.e. attribute dimension) trials and $p=0.70$ probability, and perform Hadamard product with attribute vectors of some randomly selected nodes. Thus, approximately 30% of the attributes for each selected node are corrupted. We also vary the percentage of nodes corrupted from 0%-50% to investigate the micro- and macro-F1 decline. Since we are interested

Table 4: Node classification performance in Citeseer with attribute noise.

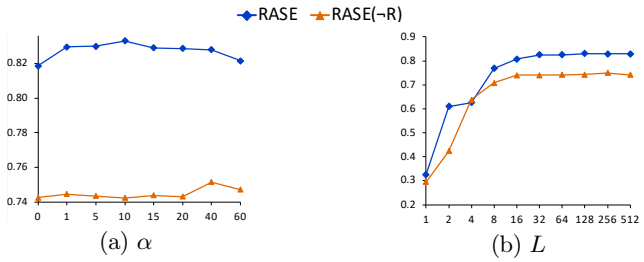
% of nodes corrupted (30% noise)	micro (mi)- and macro(ma)- F1 score											
	0%		10%		20%		30%		40%		50%	
	mi	ma	mi	ma	mi	ma	mi	ma	mi	ma	mi	ma
GraphSAGE	42.88	13.18	42.28	14.76	41.82	13.31	41.67	16.24	41.56	16.24	41.25	16.15
VGAE	77.85	76.98	76.95	77.04	76.11	76.09	75.23	75.31	74.04	74.15	73.81	73.68
G2G	84.05	84.15	82.30	82.42	81.45	81.48	79.79	79.89	78.96	78.93	79.94	79.95
RASE($\neg R$)	82.04	82.15	81.23	81.40	80.22	80.45	79.94	79.93	78.12	78.24	77.67	77.61
RASE	**85.82**	**85.78**	**84.95**	**84.96**	**83.84**	**83.80**	**83.51**	**83.44**	**82.52**	**82.39**	**82.98**	**82.85**

in evaluating the attribute robustness of the embedding methods, we experiment with attributed embedding methods only. The results are reported in Table 4.

Results in Table 4 show that RASE is robust to random node attribute noise, having the highest macro- and micro-F1 scores steadily across all noisy node percentages. Moreover, RASE only shows a 3.3% degradation in micro-F1, when we increase the proportion of corrupted nodes from 0% to 50%. The small degradation can be attributed to the node attribute denoising step. RASE also outperforms its non-robust counterpart across all the settings, showing the effectiveness of attribute noise modelling component. GraphSAGE shows a poorer node classification performance when compared to others, which shows that noisy attributes has misled the model to learn inexact node embeddings. Negative effect of noisy attributes of neighbouring nodes in GCN's aggregation step causes the lower performance of VGAE when its performance is compared against RASE and G2G. Overall, G2G shows a modest micro-F1 decline (i.e. 4.9% from 0% to 50%), since the Gaussian node embeddings have captured the attribute uncertainty via the variance terms.

4.7 Parameter Sensitivity Analysis

We study the sensitivity of attribute reconstruction learning weight (α) and embedding dimension (L) in RASE. Fig. 3 shows the micro-F1 for node classification on Citeseer averaged over 5 trials. In general, $\alpha > 0$ shows better performance than $\alpha = 0$, demonstrating the positive effect of learning from node attributes. The impact of attribute preservation is optimal near $\alpha = 10$ in RASE and $\alpha = 40$ in RASE($\neg R$). Also, RASE performs increasingly better when the embedding dimension L is increased,



(a) α (b) L

Fig. 3: Parameter Sensitivity Analysis. Micro-F1 (y-axis) in node classification on Citeseer.

since larger dimensions can encode more meaningful latent information. When $L \geq 32$, RASE and RASE($\neg R$) are already complex enough to handle the data and further increments are less helpful.

5 Conclusion and Future Work

In this work, we present RASE, an end-to-end embedding framework for attributed graphs. RASE learns robust node embeddings by preserving both graph structure and node attributes and considering random structure and attribute noise. RASE has been evaluated with respect to a number of state-of-the-art embedding methods on several public datasets in different graph analysis tasks, and the results demonstrate that our method significantly outperforms all the evaluated baselines on all the tasks. We intend to study the effect of adversarial attacks on graphs for future work.

References

1. Bojchevski, A., Günnemann, S.: Deep Gaussian Embedding of Attributed Graphs: Unsupervised Inductive Learning via Ranking. In: ICLR (2018)
2. Gao, H., Huang, H.: Deep Attributed Network Embedding. In: IJCAI (2018)
3. Givens, C.R., Shortt, R.M., et al.: A class of Wasserstein metrics for probability distributions. The Michigan Mathematical Journal **31**(2), 231–240 (1984)
4. Grover, A., Leskovec, J.: node2vec: Scalable Feature Learning for Networks. In: ACM SIGKDD (2016)
5. Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive Representation Learning on Large Graphs. In: NIPS (2017)
6. Huang, X., Li, J., Hu, X.: Label Informed Attributed Network Embedding. In: ACM WSDM (2017)
7. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: ICLR (2014)
8. Kipf, T.N., Welling, M.: Variational Graph Auto-Encoders. In: NIPS Workshop on Bayesian Deep Learning (2016)
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: NIPS (2013)
10. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: ACM SIGKDD (2014)
11. Shi, B., Weninger, T.: Open-World Knowledge Graph Completion. In: AAAI (2018)
12. Sun, G., Zhang, X.: A Novel Framework for Node/Edge Attributed Graph Embedding. In: PAKDD (2019)
13. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: Large-scale Information Network Embedding. In: WWW (2015)
14. Vilnis, L., McCallum, A.: Word Representations via Gaussian Embedding. In: ICLR (2015)
15. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.Y.: Network Representation Learning with Rich Text Information. In: IJCAI (2015)
16. Zhang, D., Yin, J., Zhu, X., Zhang, C.: SINE: Scalable Incomplete Network Embedding. In: IEEE ICDM. pp. 737–746 (2018)
17. Zhu, D., Cui, P., Wang, D., Zhu, W.: Deep Variational Network Embedding in Wasserstein Space. In: ACM SIGKDD (2018)
18. Zhu, D., Dai, X., Yang, K., Chen, J., He, Y.: PCANE: Preserving Context Attributes for Network Embedding. In: PAKDD (2019)