

Learning Stochastic Equivalence based on Discrete Ricci Curvature

Xuan Guo¹, Qiang Tian¹, Wang Zhang¹, Wenjun Wang¹ and Pengfei Jiao^{1,2*}

¹College of Intelligence and Computing, Tianjin University, Tianjin, China

²Center of Biosafety Research and Strategy, Law School, Tianjin University, Tianjin, China

{guoxuan, tianqiang, wangzhang, wjwang, pjiao}@tju.edu.cn

Abstract

Role-based network embedding methods aim to preserve node-centric connectivity patterns, which are expressions of node roles, into low-dimensional vectors. However, almost all the existing methods are designed for capturing a relaxation of automorphic equivalence or regular equivalence. They may be good at structure identification but could show poorer performance on role identification. Because automorphic equivalence and regular equivalence strictly tie the role of a node to the identities of all its neighbors. To mitigate this problem, we construct a framework called Curvature-based Network Embedding with Stochastic Equivalence (*CNESE*) to embed stochastic equivalence. More specifically, we estimate the role distribution of nodes based on discrete Ricci curvature for its excellent ability to concisely representing local topology. We use a Variational Auto-Encoder to generate embeddings while a degree-guided regularizer and a contrastive learning regularizer are leveraged to improving both its robustness and discrimination ability. The effectiveness of our proposed *CNESE* is demonstrated by extensive experiments on real-world networks.

1 Introduction

Role is a sociological concept at first, and is introduced into network science later to indicating the functions or behaviors of nodes in a network [Zhao *et al.*, 2013]. Learning roles has crucial importance for network mining and analysis, since roles have great potential for shedding light on the underlying characteristics of nodes and the evolution mechanism of networks. Therefore, role has gradually shown its importance on a number of tasks such as node classification [Ribeiro *et al.*, 2017], link prediction [Ahmed *et al.*, 2019] and visualization [Rossi *et al.*, 2018].

It is believed that nodes in the same role have highly similar node-centric connectivity patterns [Rossi and Ahmed, 2014]. A simple example is shown in Fig. 1, in which the nodes are partitioned into different roles noted by their colors. In this

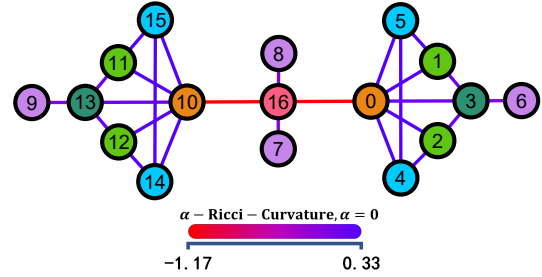


Figure 1: An example of roles in a network. The color of nodes denotes their structural traits, which determine their roles. And the color of edges denotes their α -Ricci-curvature.

example, nodes 6, 7, 8, and 9 are clustered into the same roles with only 1-hop neighborhood structures considered. These nodes are all star-edge-like, but they are significantly different when more hops of neighborhoods are considered. To practically capture higher-order node-centric structural similarity, early graph-based methods defined and leveraged different role equivalences including **structural equivalence** [Lorrain and White, 1971], **automorphic equivalence** [Holland and Leinhardt, 1981], **regular equivalence** [White and Reitz, 1983] and **stochastic equivalence** [Holland and Leinhardt, 1981; Nowicki and Snijders, 2001]. In specific, two nodes are structurally equivalent if they are connected to the same neighbors. Obviously, structural equivalence is too strict and only effective for nodes very close to each other (For example, nodes 1 and 2 in Fig. 1 are assigned into a role while nodes 11 and 12 are assigned into another role). Automorphic equivalence is a relaxation of structural equivalence as it is based on whether two nodes' node-centric subgraphs are isomorphic. Regular equivalence is relaxed further, as it states that nodes in the same role are connected to role-equivalent neighbors. Thus, regular equivalence is often used through recursive approximate manners [Jin *et al.*, 2011]. Both automorphic equivalence and regular equivalence can be used to distinguish node 6, 9 and node 7, 8 in Fig. 1. However, they make the role of each node strictly tied to neighbors' identities and roles, which leads to low generalization ability. Stochastic equivalence states that nodes in the same role have the same probability distribution of edges or node roles in their neighborhoods. With the distributions, stochastic equiv-

*Contact Author

alence reveals more essential structural traits of nodes and is more robust to noises than the other equivalences.

Recently, roles have been getting much more attention in the field of network embedding, as those classic methods only preserve proximity [Perozzi *et al.*, 2014; Grover and Leskovec, 2016] and can't work well for role/structure-based tasks. Though not all the role-based network embedding algorithms explicitly define role equivalence, most of their methods of acquiring structural similarity can be analogized to the above role equivalences. Some methods capture structural similarity via higher-order structural features. For example, RoIX [Henderson *et al.*, 2012] leverages a feature extraction method that recursively aggregates simple structural features of neighbors for each node. Struc2vec [Ribeiro *et al.*, 2017] biases random walks by constructing a hierarchical network in which the weight of edges are transformed from the degree-based distances between nodes. Role2vec [Ahmed *et al.*, 2019] uses motif-based features instead of node ids to generate sequences as input of Skip-Gram [Mikolov *et al.*, 2013]. Since these methods calculate similarity and map it into embedding space in different ways, it could be considered that they can capture a relaxation automorphic equivalence in different degrees. Besides the aforementioned methods, a few methods that leverage graph kernels to measuring structural similarities between node-centric subgraphs [Ma *et al.*, 2019; Nikolentzos and Vazirgiannis, 2019] can also be regarded as implements of automorphic equivalence. There are also some methods capturing regular equivalence. DRNE [Tu *et al.*, 2018] proposes a layer normalized LSTM to learn regular equivalence by recursive aggregating neighbors' representations for each node. Struc2gauss [Pei *et al.*, 2020] applies RoleSim [Jin *et al.*, 2011], a metric method satisfies Axiomatic Role Similarity Properties and regular equivalence. In summary, almost all the network embedding methods could achieve capturing role equivalences that are stricter than stochastic equivalence and more relaxed than the automorphic equivalence. As far as we know, there are so few previous network embedding methods satisfying a stochastic equivalence criterion, which could be more flexible and robust for role-based tasks.

Therefore, we introduce stochastic equivalence into network embedding algorithms. The key is to choose a proper representation method of edges, which should satisfy the following properties: (1) The representation of an edge can concisely generalize its local topology. (2) The representations are easy to utilize for learning latent role distributions of nodes. To meet above two requirements, we choose discrete Ricci curvature [Chung and Yau, 1996; Ollivier, 2009; Lin *et al.*, 2011] to represent edges. Ricci curvature of an edge is a real value that measures its importance to the closeness of the two neighborhoods. Thus, Ricci curvature can be used to distinguish edge roles such as near-clique edges, bridge edges, and star edges. Specifically, we leverage α -Ricci-curvature [Lin *et al.*, 2011] (as denoted by edge color in Fig. 1) for it can not only generalize local topology but also use a balance parameter α to control the influence of neighbors. To further take advantages of α -Ricci-curvature, we use a Variational Auto-Encoder (VAE) [Kingma and Welling, 2014] to learn the latent representations of role dis-

tributions for each node while a contrastive learning regularizer and a degree-guided regularizer [Tu *et al.*, 2018; Zhang *et al.*, 2021] are applied to enhance its robustness and ability of discrimination. Therefore, we can preserve the stochastic equivalence into embedding space with this framework called **Curvature-based Network Embedding with Stochastic Equivalence (CNESE)**¹.

Our main contributions can be summarized as follows:

- We first explicitly propose the idea of bestowing network embedding with the ability to capture stochastic equivalence. We are also the first to introduce discrete Ricci curvature to capture stochastic equivalence and elucidate why discrete Ricci curvature works.
- We propose a framework CNESE based on a Variational Auto-Encoder regularized by both a contrastive learning regularizer and a degree-guided regularizer. It can effectively leverage α -Ricci-curvature to generating high-quality role-based embeddings.
- Extensive experiments on real-world datasets demonstrate that our proposed method CNESE achieves state-of-the-art performance.

2 Preliminaries

2.1 Notations and Problem Definition

An unweighted undirected network is usually represented as a graph $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$ is the set of nodes and $E \subseteq V \times V$ is the set of edges. For each node $v \in V$, the set of its neighbors is represented as $\Gamma(v) = \{u | (v, u) \in E\}$ and its degree is $k_v = |\Gamma(v)|$.

Role-based network embedding aims to learning a function mapping nodes in G to into a low-dimensional space, denoted as $f : V \rightarrow \mathcal{Z} \in \mathbb{R}^{n \times d}$, where \mathcal{Z}_i is the embedding of node v_i preserving its node-centric connectivity patterns.

2.2 Discrete Ricci Curvature on Graphs

In this paper, we use the notion of coarse Ricci curvature introduced by Ollivier [Ollivier, 2009]. Given probability measures m_u and $m_v : V \rightarrow [0, 1]$ for nodes $u, v \in V$ with mass 1 respectively, a mass-preserving transportation plan between them $A : V \times V \rightarrow [0, 1]$ meets $\sum_{y \in V} A(x, y) = m_u(x)$ and $\sum_{x \in V} A(x, y) = m_v(y)$. The transportation distance between the two probability distributions m_u and m_v is defined as the Wasserstein distance between them, i.e., the minimum average traveling distance via any transportation plan:

$$W(m_u, m_v) = \inf_A \sum_{x, y \in V} A(x, y) d(x, y), \quad (1)$$

where $d(x, y)$ is the graph distance between x and y . Then Ollivier's coarse Ricci curvature for any two nodes $u, v \in V$ is defined as:

$$\kappa(u, v) = 1 - \frac{W(m_u, m_v)}{d(u, v)}. \quad (2)$$

By comparing the transportation distance and the graph distance between node u and v , Ricci curvature measures how

¹<https://github.com/cspjiao/CNESE>

important the shortest path between the two nodes is for making them closer.

We follow [Lin *et al.*, 2011] to define the probability distribution for node $v \in V$ with a parameter $\alpha \in [0, 1]$:

$$m_v^\alpha(x) = \begin{cases} \alpha & \text{if } x = v \\ (1 - \alpha)/k_v & \text{if } x \in \Gamma(v) \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Through (3) we can define α -Ricci-curvature for any $u, v \in V$ as $\kappa_\alpha(u, v) = 1 - W(m_u^\alpha, m_v^\alpha)/d(u, v)$. We can tune α to balance the influence of node v itself and its neighborhood. It make sense for real-world networks as the role of a node depends on both itself and its relations to neighbors.

For an edge in G , α -Ricci-curvature can measure not only its importance for the closeness between the endpoints' neighborhoods but also the endpoints' degrees. α -Ricci-curvature not only concisely generalizes the local topology of an edge, but also allows for easy estimation of the role distributions as a real value. Thus, we leverage it to capturing stochastic equivalence.

3 Method

In this section, our proposed framework Curvature-based Network Embedding with Stochastic Equivalence (CNESE) is introduced. An overview of CNESE is illustrated in Fig. 2.

3.1 Computation of Ricci Curvature

We efficiently and effectively compute $W(m_u^\alpha, m_v^\alpha)$ via linear programming as in [Ni *et al.*, 2015]:

$$\begin{aligned} \min \quad & \sum_{y \in V} \sum_{x \in V} d(x, y) \rho_{x, y} m_u^\alpha(x), \\ \text{s.t.} \quad & 0 \leq \rho_{xy} \leq 1 \quad \forall x, y \in V, \\ & \sum_{y \in V} \rho_{xy} = 1 \quad \forall x \in V, \\ & \sum_{x \in V} \rho_{xy} m_u^\alpha(x) = m_v^\alpha(y) \quad \forall y \in V, \end{aligned} \quad (4)$$

where the matrix ρ represents the transportation plan. Then, we can obtain the α -Ricci-curvature for all the edges in G . To capture stochastic equivalence, we estimate the distributions of Ricci curvatures using a fuzzy but effective method, histogram, on the connected edges for all the nodes.

3.2 Variational Auto-Encoder

To preserve stochastic equivalence into low-dimensional vector space, we utilize a Variational Auto-Encoder through reconstructing Ricci curvature histograms.

We assume that the latent representation \mathbf{Z}_i of node $v_i \in V$ is an d -dimensional independent random vector matching a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. The diagonal entries of $\boldsymbol{\Sigma}_i$ is represented as a vector $\boldsymbol{\sigma}_i$. Then we generate \mathbf{Z}_i as follows:

$$\begin{aligned} p(\mathbf{Z}_i | \mathbf{H}_i) &= \prod_{j=1}^d p(\mathbf{Z}_{ij} | \mathbf{H}_i), \\ p(\mathbf{Z}_{ij} | \mathbf{H}_i) &= \mathcal{N}(\mathbf{Z}_{ij} | \boldsymbol{\mu}_{ij}, \boldsymbol{\sigma}_{ij}^2), \end{aligned} \quad (5)$$

where \mathbf{H} is the Ricci curvature histogram matrix for all the nodes.

The mean vector $\boldsymbol{\mu}_i$ and variance vector $\boldsymbol{\sigma}_i$ are learned as follows:

$$\begin{aligned} \mathbf{Z}'_i &= \text{MLP}_{\text{enc1}}(\mathbf{H}_i), \\ \boldsymbol{\mu}_i &= \text{MLP}_{\text{enc2}}(\mathbf{Z}'_i), \\ \log \boldsymbol{\sigma}_i &= \text{MLP}_{\text{enc3}}(\mathbf{Z}'_i). \end{aligned} \quad (6)$$

Here, MLP means a multi-layer perceptron model. Thus our variational encoder $\mathcal{G}(\cdot)$ is composed of (5) and (6).

We also use a multi-layer perceptron as our decoder to reconstruct \mathbf{H} :

$$\hat{\mathbf{H}}_i = \text{MLP}_{\text{dec}}(\mathbf{Z}_i). \quad (7)$$

The loss of reconstruction is:

$$\mathcal{L}_{\text{vae}} = \left\| \hat{\mathbf{H}} - \mathbf{H} \right\|_F^2. \quad (8)$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

3.3 Degree-Guided Regularizer

The Ricci curvature histograms could be a double-edged sword on embedding generation sometimes. Though the fuzziness of histograms makes estimating stochastic equivalence easier, it could abate the structure information contained in Ricci curvatures. This problem is particularly prone to occur on very small or sparse networks. To mitigate this problem, we leverage a degree-guided regularizer as in DRNE [Tu *et al.*, 2018]. In detail, we use a multi-layer perceptron as an Approximator, i.e., $\mathcal{A}(\cdot) = \text{MLP}_{\text{app}}(\cdot)$, to approximate the degree of each node. Our degree-guided regularizer is designed as follows:

$$\mathcal{L}_{\text{deg}} = \sum_{i=1}^n (\log(k_{v_i} + 1) - \mathcal{A}(\mathcal{G}(\mathbf{H}_i)))^2. \quad (9)$$

Thus, the degree-guided regularizer can bestow the embeddings with guidance information of degrees.

3.4 Contrastive Learning Regularizer

To generate more robust embeddings, we construct a contrastive learning regularizer for our framework. We build a Discriminator using a multi-layer perceptron, i.e., $\mathcal{D}(\cdot) = \text{MLP}_{\text{dis}}(\cdot)$, which is essentially a binary classifier for distinguishing negative samples \mathbf{Z}^- and latent representations generated by $\mathcal{G}(\mathbf{H}_i)$. As many negative samples are sampled from the standard Gaussian distribution $q(\mathbf{Z}_j^-) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ as positive samples (node embeddings). The objective of the contrastive learning regularizer is as follows:

$$\begin{aligned} \mathcal{L}_{\text{con}} &= \frac{1}{2n} \left(\sum_{i=1}^n \mathbb{E}_{\mathbf{Z}^-} \log(\mathcal{D}(\mathbf{Z}_i^-)) \right. \\ &\quad \left. + \sum_{j=1}^n \mathbb{E}_{\mathbf{H}} \log(1 - \mathcal{D}(\mathcal{G}(\mathbf{H}_j))) \right). \end{aligned} \quad (10)$$

The Discriminator $\mathcal{D}(\cdot)$ is trained as follows:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}_{\text{con}}. \quad (11)$$

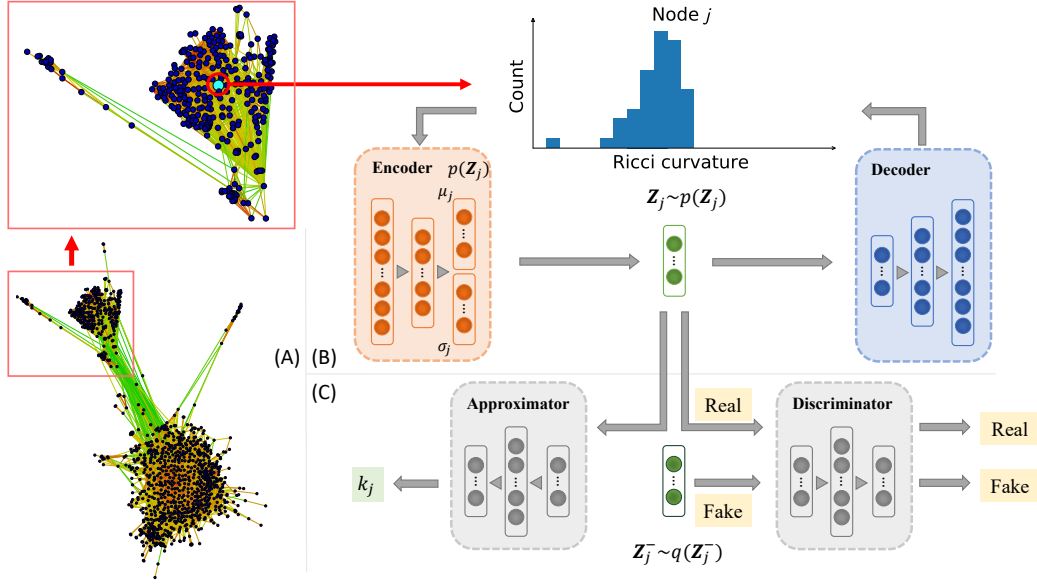


Figure 2: An overview of Curvature-based Network Embedding with Stochastic Equivalence (CNESE). (A) Compute Ricci curvature for all the edges and represent each node with a Ricci curvature histogram on all the edges connected to it. (B) Generate embeddings via a Variational Auto-Encoder. (C) Enhance the embeddings with a contrastive learning regularizer and a degree-guided regularizer.

Dataset	# nodes	# edges	# classes
Brazil	131	1,003	4
Enron	143	2,583	7
Europe	399	5,993	4
USA	1,190	13,599	4
Reality	6,809	7,680	3
Actor	7,758	26,646	4

Dataset	# nodes	# edges	# bots
br-Wiki-talk	1,049	2,330	35
cy-Wiki-talk	2,101	3,610	31

Table 1: Statistics of datasets.

In total, we embed Ricci curvature histogram matrix H into a low-dimensional vector space by minimizing a combined loss function as follows:

$$\mathcal{L} = \mathcal{L}_{vae} + \beta \mathcal{L}_{deg} + \gamma \mathcal{L}_{con}. \quad (12)$$

where β and γ are the weights of the degree-guided regularizer and the contrastive learning regularizer.

4 Experiments

In this section, we want to validate the role discrimination ability of our CNESE through experiments on role-based node classification, bot detection and visualization.

4.1 Experimental Setup

Datasets and Baselines

We conduct the role-based node classification experiments on Air-traffic networks (**USA**, **Brazil**, and **Europe**) [Ribeiro *et al.*

MLP	layer number	activation functions
MLP _{enc1}	2	$\tanh(\cdot)$
MLP _{enc2}	1	— [†]
MLP _{enc3}	1	—
MLP _{dec}	3	$\tanh(\cdot)$, — [‡]
MLP _{app}	2	$\text{ReLU}(\cdot)$
MLP _{dis}	3	$\text{ReLU}(\cdot)$, $\text{sigmoid}(\cdot)$

[†] — means no activation function is applied.

[‡] The second activation function is applied on the last layer, while the first is applied on the other layers.

Table 2: Configuration of multi-layer perceptrons.

al., 2017], **Actor** co-occurrence network [Ma *et al.*, 2019], **Reality** phone call network [Guo *et al.*, 2020], and **Enron** email network [Klimt and Yang, 2004]. We also do an experiment to detect bot users in two Wikipedia user interaction networks (**br-Wiki-talk** and **cy-Wiki-talk**) [Sun *et al.*, 2016]. Some statistics of these datasets are provided in Table 1.

We compare our CNESE with a set of baselines including classic (community-based) and the state-of-the-art role-based network embedding methods. For classic methods, we use DeepWalk [Perozzi *et al.*, 2014] and node2vec [Grover and Leskovec, 2016]. For role-based methods, we use RolX [Henderson *et al.*, 2012], struc2vec [Ribeiro *et al.*, 2017], GraphWave [Donnat *et al.*, 2018], DRNE [Tu *et al.*, 2018] and struc2gauss [Pei *et al.*, 2020]. The parameters of all these methods are well tuned for the best performance.

Model Configuration

MLPs with different layer numbers and different activation functions are applied on different parts of CNESE accord-

Method	Brazil		Enron		Europe		USA		Reality		Actor	
	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
DeepWalk	41.45	39.94	29.73	19.96	37.77	37.01	52.46	51.81	49.71	32.40	36.09	34.73
node2vec	40.50	40.09	28.43	20.30	40.50	40.09	50.02	49.63	50.98	33.32	35.60	34.56
RoIX	71.07	70.53	31.77	18.71	54.95	53.64	64.24	63.52	65.38	42.90	46.63	45.23
struc2vec	72.62	72.05	28.20	17.17	57.88	57.52	61.13	60.36	60.74	38.74	46.79	46.11
GraphWave	73.00	71.44	36.17	19.07	53.31	49.53	51.80	47.30	83.55	51.80	OM	OM
DRNE	77.58	76.87	31.43	16.42	54.14	52.38	59.31	57.95	83.89	52.44	48.27	45.90
struc2gauss	74.60	73.24	31.00	16.85	55.17	55.10	62.83	62.22	60.32	39.29	44.95	44.18
<i>CNESE</i>	77.37	76.37	36.33	22.26	60.48	60.13	63.18	61.96	84.02	51.61	48.80	47.99

To save space, Micro-F1 and Macro-F1 are shortly denoted as Micro and Macro respectively. OM indicates out-of-memory error.

Table 3: Node classification average F1 score (%) on different networks.

ing to their different purposes, as shown in TABLE 2. The bin number of curvature histograms is set to 80. The width of MLPs and embedding dimension are set to 64. We apply Adam SGD optimizer [Kingma and Ba, 2015] with learning rate 0.001 and batch size 32 for at most 50 epochs. L_2 regularization with weight 0.001 is adopted to avoid overfitting. In later experiments, unless otherwise stated, parameter α of α -Ricci-curvature is set as 0.05, and 0.5 on Brazil and the other datasets respectively. β is set to 5 and γ is set to 2.

4.2 Role-based Node Classification

We conduct node classification tasks on 6 datasets: Brazil, Enron, Europe, USA, Reality, and Actor. Specifically, a linear logistic regression classifier is trained and tested on embeddings for each dataset. We sample 70% node embeddings as a training set and the other embeddings are used as the test set. We run all the baselines and *CNESE* 25 times on each network and calculate the average of micro-averaged F1 score (Micro-F1) and macro-averaged F1 score (Macro-F1) for every method. We report the results in TABLE 3 and the best results are indicated in bold.

From these results, we can observe that all the role-based network embedding methods outperform classic methods on almost all the networks except for Enron network. On most datasets, results of role-based methods are at least 10% higher than those of DeepWalk and node2vec. This demonstrates that methods preserving node proximity are not appropriate to role-based node classification tasks.

Our proposed *CNESE* achieves better performance than the other role-based methods on most datasets. *CNESE* performs very closely to the best results in the cases where *CNESE* is not the top. This demonstrates the stability and effectiveness of our *CNESE*. On Brazil dataset, DRNE gives the best results, because the network is too small to estimate the equivalence between distributions while node labels and node degrees are highly correlated. RoIX yields the best results on USA network as the roles are more similar to automorphic equivalence. On Enron network, DeepWalk and node2vec perform closely to some role-based methods. This is because node labels in Enron denote real employee titles unlike those of the other networks based on specific function measures. In this case, methods capturing automorphic equivalence and regular equivalence lose effectiveness while *CNESE* and GraphWave still perform greatly. Thus,

Method	br-Wiki-talk		cy-Wiki-talk	
	P@5	P@25	P@5	P@25
DeepWalk	0.0629	0.0491	0.1290	0.0852
node2vec	0.0286	0.0423	0.0581	0.0645
RoIX	0.2171	0.1291	0.0968	0.1019
struc2vec	0.1543	0.1257	0.1097	0.0981
GraphWave	0.2114	0.1211	0.1484	0.1084
DRNE	0.1714	0.1371	0.1355	0.0929
struc2gauss	0.1886	0.1543	0.1484	0.1123
<i>CNESE</i>	0.2229	0.1577	0.1613	0.1213

Table 4: Precision at K selected users (P@K) on bot detection task.

our *CNESE* shows its wide applicability.

4.3 Bot Detection

In this section, we investigate whether our *CNESE* is able to find the nodes playing rare roles. We carry out a bot detection experiment on the two Wikipedia user interaction networks. First, we exert network embedding methods to generate representations for all the users. Then we calculate the euclidean distance between users and take out K closest users for each bot. Finally, we compute the P@K, i.e., the average precision on the selected K users for all bots. The results are given in TABLE 4 and the best results are indicated in bold.

On br-Wiki-talk network, all results of role-based methods are at least twice as good as those of DeepWalk and node2vec. Our results are even three times as precise as theirs. These results demonstrate that bots in br-Wiki-talk network are strongly structurally similar to each other and our *CNESE* is better at capturing this structural similarity. As for cy-Wiki-talk network, the results of DeepWalk and node2vec improve while those of role-based methods deteriorate. Some role-based methods perform worse than DeepWalk and node2vec but *CNESE* still achieves the best results. It shows that bots in cy-Wiki-talk network are less structurally identical and our *CNESE* have favorable adaptability on the task with weaker structural characteristics.

4.4 Parameter Sensitivity Analysis

Here, we investigate the influence of parameters including the balance parameter of α -Ricci-curvature α , the weights of two regularizers β and γ , and the dimension of embeddings d . We conduct role-based node classification on Brazil, Eu-

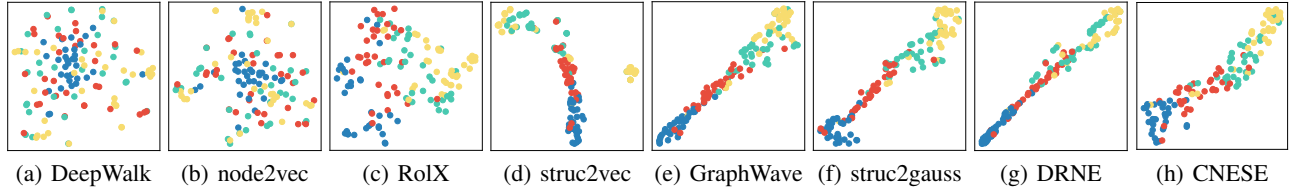
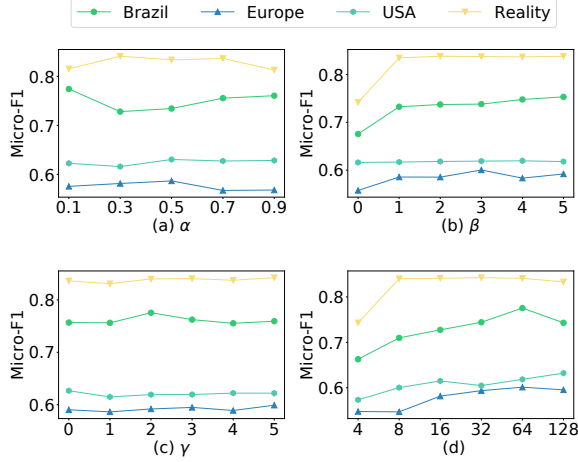


Figure 3: t-SNE visualization of embeddings on Brazil network. The node labels are indicated by their colors.


 Figure 4: Parameter sensitivity on several networks with respect to (a) the balance parameter of α -Ricci-curvature α , (b) the weight of degree-guided regularizer β , (c) the weight of contrastive learning regularizer γ and (d) the dimension of embeddings d .

rope, USA, and Reality networks and report only the Micro-F1 score in Fig. 4 for brevity. We set $\alpha = 0.5$ ($\alpha = 0.05$ on Brazil), $\beta = 1$, $\gamma = 1$, and $d = 64$ when they are control variables. In Fig. 4(a), the performance deteriorates firstly and then raises slightly when α increases on Brazil. The opposite trend is shown on other networks. This indicates that nodes express their functions not only on the relations to their neighbors but also on themselves. As shown in Fig. 4(b) and Fig. 4(c), we illustrate the influence of the weights of two regularizers β and γ . As β gets larger, the performance of *CNESE* gets better, while on Brazil, it gets the best performance at $\beta = 3$. It demonstrates that node degree can help capture stochastic equivalence, but when it is large enough, the original information may lose. In Fig. 4(c), the performance jumps up to the best at $\gamma = 2$ and drops when γ is larger on Brazil network. The reason is that contrastive learning with negative samples can prevent *CNESE* learning wrong distributions. It may mislead the learning process on small size networks like Brazil, since too few positive samples can be used for training. And for embedding dimension d shown in Fig. 4(d), the performance raises when d increases, and we can obtain good enough result with $d = 64$.

We also take ablation analysis of the contrastive learning and degree-guided regularizer. In Fig. 4(b), when the parameter β is set to 0, it denotes the model *CNESE* without regularizer of node degree. We observe a noticeable increase

when this regularizer is added to our model, which means that degree has a significant contribution to the role of nodes, and the process of extracting features and encoding loses the information. As for the part of contrastive learning, we report the effectiveness in Fig. 4(c) when $\gamma = 0$. In this situation, the model only consists of VAE and degree-guided regularizer, and we find that the performance decreases firstly but gets best when $\gamma = 2$.

4.5 Visualization

In this section, we compared our *CNESE* with baselines on visualization tasks. We apply t-SNE to reduce the dimension of embeddings learned by different methods. All the embeddings on Brazil network are mapped as two-dimensional vectors. The results are illustrated in Fig. 3. Each node is represented by a point while the node label is denoted by the color of the corresponding point. Ideally, the points in the same color should be close to each other and those in different colors should be far from each other. From the subfigures of DeepWalk and node2vec, it can be observed that the points in four colors are mixed together. In the subfigures of role-based methods, the points form areas based on color. Furthermore, GraphWave, DRNE, struc2gauss, and our *CNESE* achieve more ideal results. However, the points generated by GraphWave, DRNE, and struc2gauss are almost lined up. This indicates that these methods may overfit only one structural property. For example, DRNE could be over-dependent on degree guidance. Though we leverage a degree-guided regularizer to *CNESE* as well, our method can contain more structure information as Ricci curvature does.

5 Conclusion

In this paper, we provide and explain the idea of preserving stochastic equivalence into network embeddings. To implement the idea, we leverage Ricci curvature to generalize local topology and explain why it works. Then we construct a role-based embedding framework *CNESE* that combines Variational Auto-Encoder with a degree-guided regularizer and a contrastive learning regularizer and treat Ricci curvature histograms as input. Finally, we verify the effectiveness of *CNESE* with extensive experiments. Though we design our *CNESE* for undirected networks, it is easy to extend it on directed networks.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61902278).

References

- [Ahmed *et al.*, 2019] Nesreen K Ahmed, Ryan A Rossi, John Boaz Lee, Theodore L Willke, Rong Zhou, Xiangnan Kong, and Hoda Eldardiry. role2vec: Role-based network embeddings. In *Proc. DLG KDD*, 2019.
- [Chung and Yau, 1996] Fan RK Chung and Shing-Tung Yau. Logarithmic harnack inequalities. *Mathematical Research Letters*, 3(6):793–812, 1996.
- [Donnat *et al.*, 2018] Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. Learning structural node embeddings via diffusion wavelets. In *SIGKDD*, pages 1320–1329, 2018.
- [Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, pages 855–864, 2016.
- [Guo *et al.*, 2020] Xuan Guo, Wang Zhang, Wenjun Wang, Yang Yu, Yinghui Wang, and Pengfei Jiao. Role-oriented graph auto-encoder guided by structural information. In *DASFAA*, pages 466–481. Springer, 2020.
- [Henderson *et al.*, 2012] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. Rolx: structural role extraction & mining in large graphs. In *SIGKDD*, pages 1231–1239, 2012.
- [Holland and Leinhardt, 1981] Paul W Holland and Samuel Leinhardt. An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50, 1981.
- [Jin *et al.*, 2011] Ruoming Jin, Victor E Lee, and Hui Hong. Axiomatic ranking of network role similarity. In *SIGKDD*, pages 922–930, 2011.
- [Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.
- [Klimt and Yang, 2004] Bryan Klimt and Yiming Yang. Introducing the enron corpus. In *CEAS*, 2004.
- [Lin *et al.*, 2011] Yong Lin, Linyuan Lu, and Shing-Tung Yau. Ricci curvature of graphs. *Tohoku Mathematical Journal, Second Series*, 63(4):605–627, 2011.
- [Lorrain and White, 1971] Francois Lorrain and Harrison C White. Structural equivalence of individuals in social networks. *The Journal of mathematical sociology*, 1(1):49–80, 1971.
- [Ma *et al.*, 2019] Xuwei Ma, Geng Qin, Zhiyang Qiu, Mingxin Zheng, and Zhe Wang. Riwalk: Fast structural node embedding via role identification. In *ICDM*, pages 478–487, 2019.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.
- [Ni *et al.*, 2015] Chien-Chun Ni, Yu-Yao Lin, Jie Gao, Xian-feng David Gu, and Emil Saucan. Ricci curvature of the internet topology. In *INFOCOM*, pages 2758–2766. IEEE, 2015.
- [Nikolentzos and Vazirgiannis, 2019] Giannis Nikolentzos and Michalis Vazirgiannis. Learning structural node representations using graph kernels. *IEEE Transactions on Knowledge and Data Engineering*, 2019.
- [Nowicki and Snijders, 2001] Krzysztof Nowicki and Tom A B Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American statistical association*, 96(455):1077–1087, 2001.
- [Ollivier, 2009] Yann Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864, 2009.
- [Pei *et al.*, 2020] Yulong Pei, Xin Du, Jianpeng Zhang, George Fletcher, and Mykola Pechenizkiy. struc2gauss: Structural role preserving network embedding via Gaussian embedding. *Data Mining and Knowledge Discovery*, 34(4):1072–1103, 2020.
- [Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD*, pages 701–710, 2014.
- [Ribeiro *et al.*, 2017] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *SIGKDD*, pages 385–394, 2017.
- [Rossi and Ahmed, 2014] Ryan A Rossi and Nesreen K Ahmed. Role discovery in networks. *IEEE Transactions on Knowledge and Data Engineering*, 27(4):1112–1131, 2014.
- [Rossi *et al.*, 2018] Ryan A Rossi, Nesreen K Ahmed, Rong Zhou, and Hoda Eldardiry. Interactive visual graph mining and learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(5):1–25, 2018.
- [Sun *et al.*, 2016] Jun Sun, Jérôme Kunegis, and Steffen Staab. Predicting user roles in social networks using transfer learning with feature transformation. In *ICDMW*, pages 128–135. IEEE, 2016.
- [Tu *et al.*, 2018] Ke Tu, Peng Cui, Xiao Wang, Philip S Yu, and Wenwu Zhu. Deep recursive network embedding with regular equivalence. In *SIGKDD*, pages 2357–2366, 2018.
- [White and Reitz, 1983] Douglas R White and Karl P Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5(2):193–234, 1983.
- [Zhang *et al.*, 2021] Wang Zhang, Xuan Guo, Wenjun Wang, Qiang Tian, Lin Pan, and Pengfei Jiao. Role-based network embedding via structural features reconstruction with degree-regularized constraint. *Knowledge-Based Systems*, 218:106872, 2021.
- [Zhao *et al.*, 2013] Yuchen Zhao, Guan Wang, Philip S Yu, Shaobo Liu, and Simon Zhang. Inferring social roles and statuses in social networks. In *SIGKDD*, pages 695–703, 2013.