These k-step motif-based matrices can densify quickly and therefore we recommend using $K \leq 4$. Given a k-step motif-based matrix $\mathbf{S}_t^{(k)}$ for an arbitrary network motif $H_t \in \mathcal{H}$, we learn node embeddings by solving the following objective function:

$$\underset{\mathbf{U}_t^{(k)}, \mathbf{V}_t^{(k)} \in C}{\arg\min} \; \mathbb{D}\big(\mathbf{S}_t^{(k)} \parallel \Phi\langle \mathbf{U}_t^{(k)} \mathbf{V}_t^{(k)} \rangle \big) \tag{7}$$

where $\mathbb{D}$ is a generalized Bregman divergence with matching linear or non-linear function Φ and C denotes constraints (*e.g.*, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$). We use Eq. 7 to learn a $N \times D_\ell$ local embedding $\mathbf{U}_t^{(k)}$ from $\mathbf{S}_t^{(k)}$ for all $t = 1, \ldots, T$ and $k = 1, \ldots, K$.[1] Afterwards, we scale each column of $\mathbf{U}_t^{(k)}$ using the Euclidean norm. Next, we concatenate the k-step embedding matrices for all T motifs and all K steps:

$$\mathbf{Y} = \Big[\underbrace{\mathbf{U}_1^{(1)} \cdots \mathbf{U}_T^{(1)}}_{\text{1-step}} \cdots \underbrace{\mathbf{U}_1^{(K)} \cdots \mathbf{U}_T^{(K)}}_{K\text{-steps}} \Big] \tag{8}$$

where $\mathbf{Y}$ is a $N \times TKD_\ell$ matrix. Given $\mathbf{Y}$, we learn a *global* higher-order network embedding by solving the following:

$$\underset{\mathbf{Z}, \mathbf{H} \in C}{\arg\min} \; \mathbb{D}\big(\mathbf{Y} \parallel \Phi\langle \mathbf{ZH} \rangle \big) \tag{9}$$

where $\mathbf{Z}$ is a $N \times D$ matrix of node embeddings. In Eq. 9 we use Frobenius norm which leads to the following minimization problem:

$$\min_{\mathbf{Z},\mathbf{H}} \; \frac{1}{2} \big\| \mathbf{Y} - \mathbf{ZH} \big\|_F^2 = \frac{1}{2} \sum_{ij} \big(\mathbf{Y}_{ij} - (\mathbf{ZH})_{ij} \big)^2 \tag{10}$$

A similar minimization problem is solved for Eq. 7.

3 EXPERIMENTS

We compare the proposed HONE variants to five recent state-of-the-art methods (see Table 1). All methods output ($D = 128$)-dimensional node embeddings $\mathbf{Z} = \big[\mathbf{z}_1 \cdots \mathbf{z}_N \big]^T$ where $\mathbf{z}_i \in \mathbb{R}^D$. For node2vec, we perform a grid search over $p, q \in \{0.25, 0.5, 1, 2, 4\}$ as mentioned in [4]. All other hyperparameters for node2vec [4], DeepWalk [5], and LINE [9] correspond to those mentioned in [4]. In contrast, the HONE variants have only one hyperparameter, namely, the number of steps K which is selected automatically via a grid search over $K \in \{1, 2, 3, 4\}$ using 10% of the labeled data. We use all 2-4 node connected orbits [6] and set $D_\ell = 16$ for the local motif embeddings. All methods use logistic regression (LR) with an L2 penalty. The model is selected using 10-fold cross-validation on 10% of the labeled data. Experiments are repeated for 10 random seed initializations. Data was obtained from [7].

We evaluate the HONE variants for link prediction. Given a partially observed graph G with a fraction of missing edges, the link prediction task is to predict these missing edges. We generate a labeled dataset of edges. Positive examples are obtained by removing 50% of edges randomly, whereas *negative examples* are generated by randomly sampling an equal number of node pairs $(i, j) \notin E$. For each method, we learn embeddings using the remaining graph. Using the embeddings from each method, we then learn a model to predict whether a given edge in the test set exists in E or not.

[1] For the motif Laplacian matrix formulations proposed above, we also investigated using the eigenvectors of the D_ℓ smallest eigenvalues of $\Psi(\mathbf{W}_t^k)$ as node embeddings.

Table 1: AUC results comparing HONE to recent embedding methods. See text for discussion.

	soc-hamster	*rt-twitter-cop*	*soc-wiki-Vote*	*tech-routers-rf*	*facebook-PU*	*inf-openflights*	*soc-bitcoinA*	Rank
HONE-W (Eq. 1)	0.841	0.843	0.811	0.862	0.726	0.910	0.979	**1**
HONE-P (Eq. 3)	0.840	0.840	0.812	0.863	0.724	0.913	0.980	**2**
HONE-L (Eq. 4)	0.829	0.841	0.808	0.858	0.722	0.906	0.975	**3**
HONE-$\widehat{\mathbf{L}}$ (Eq. 5)	0.829	0.836	0.803	0.862	0.722	0.908	0.976	**4**
Node2Vec [4]	0.810	0.635	0.721	0.804	0.701	0.844	0.894	**5**
DeepWalk [5]	0.796	0.621	0.710	0.796	0.696	0.837	0.863	**6**
LINE [9]	0.752	0.706	0.734	0.800	0.630	0.837	0.780	**7**
GraRep [3]	0.805	0.672	0.743	0.829	0.702	0.898	0.559	**8**
Spectral [10]	0.561	0.699	0.593	0.602	0.516	0.606	0.629	**9**

To construct edge features from the node embeddings, we use the mean operator defined as $(\mathbf{z}_i + \mathbf{z}_j)/2$. The AUC results are provided in Table 1. In all cases, the HONE methods outperform the other embedding methods with an overall mean gain of 19.24% (and up to 75.21% gain) across a wide variety of graphs with different characteristics. Overall, the HONE variants achieve an average gain of 10.68% over node2vec, 12.56% over DeepWalk, 13.79% over LINE, 17.17% over GraRep, and 41.99% over Spectral clustering across all networks. We also derive a total ranking of the embedding methods over all graph problems based on mean relative gain (1-vs-all). Results are provided in the last column of Table 1.

4 CONCLUSION

In this work, we introduced higher-order network representation learning and proposed a general framework called *higher-order network embedding* (HONE) for learning such embeddings based on higher-order connectivity patterns. The experimental results demonstrate the effectiveness of learning higher-order network representations. Future work will investigate the framework using other useful motif-based matrices.

REFERENCES

[1] Nesreen K. Ahmed, Jennifer Neville, Ryan A. Rossi, and Nick Duffield. 2015. Efficient Graphlet Counting for Large Networks. In *ICDM*. 10.

[2] Nesreen K. Ahmed, Ryan A. Rossi, Theodore L. Willke, and Rong Zhou. 2017. Edge Role Discovery via Higher-Order Structures. In *PAKDD*. 291–303.

[3] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning graph representations with global structural information. In *CIKM*. ACM, 891–900.

[4] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *SIGKDD*. 855–864.

[5] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *SIGKDD*. 701–710.

[6] Nataša Pržulj. 2007. Biological network comparison using graphlet degree distribution. *Bioinfo.* 23, 2 (2007), e177–e183.

[7] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*. 4292–4293. http://networkrepository.com

[8] Ryan A. Rossi and Nesreen K. Ahmed. 2015. Role Discovery in Networks. *Transactions on Knowledge and Data Engineering* 27, 4 (April 2015), 1112–1131.

[9] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *WWW*. 1067–1077.

[10] Lei Tang and Huan Liu. 2011. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery* 23, 3 (2011), 447–478.