Explaining Node Embeddings

Anonymous authors
Paper under double-blind review

Abstract

Node embedding algorithms produce low-dimensional latent representations of nodes in a graph. These embeddings are often used for downstream tasks, such as node classification and link prediction. In this paper, we investigate the following two questions: (Q1) Can we explain each embedding dimension with human-understandable graph features (e.g. degree, clustering coefficient and PageRank). (Q2) How can we modify existing node embedding algorithms to produce embeddings that can be easily explained by human-understandable graph features? We find that the answer to Q1 is yes and introduce a new framework called XM (short for eXplain eMbedding) to answer Q2. A key aspect of XM involves minimizing the nuclear norm of the generated explanations. We show that by minimizing the nuclear norm, we minimize the lower bound on the entropy of the generated explanations. We test XM on a variety of real-world graphs and show that XM not only preserves the performance of existing node embedding methods, but also enhances their explainability.

1 Introduction

Graph datasets are ubiquitous from social, to information, to physical, to biological networks. The abundance of graph data has inspired the development of models and algorithms in the field of graph machine learning (Chami et al., 2022), a popular task being node embedding,[1] where nodes are embedded in a low-dimensional latent space. Many algorithms exist for embedding a graph's nodes into a low-dimensional space. Zhang et al. (2021a) compared a variety of such algorithms on tasks such as greedy routing and link prediction. Recently, much attention has been devoted to explainable graph machine learning (Burkart & Huber, 2021), in part because the performance of such models can often be misleading (Stolman et al., 2022), (Menand & Seshadhri, 2024). Most works on explainable graph machine learning focus on explaining predictions for downstream tasks. In this work, we study how to assign human-understandable features to node embeddings. For example, a human-understandable embedding dimension represents nodes with many neighbors and few triangles. We also discuss why the objective functions of common node embedding algorithms are not designed to produce human-understandable node embeddings, and how they can be extended to produce such desirable node embeddings.

Recent work on explaining graph machine learning revolves around explaining decisions made on downstream tasks (Pope et al., 2019), (Baldassarre & Azizpour, 2019), (Ying et al., 2019), (Luo et al., 2020), (Vu & Thai, 2020), Yuan et al. (2021), (Liu et al., 2018)]. An outlier here is Gogoglou et al. (2019)'s work. They investigate individual dimensions of an embedding and define an interpretability score for each dimension to measure how well each dimension defines a subgroup of nodes (see Section 5 for details). In contrast, our aim is to explain each embedding dimension in terms of a user-defined set of human-understandable (a.k.a. "sense") features. A byproduct of our approach is that for each node in the graph we can explain its position in the embedding space. Given that many embedding algorithms exist, we propose a framework (called XM) to extend these existing algorithms to make their embeddings explainable. The predictive performance and runtime results of the XM variants are comparable to those of the original methods with AUC on link prediction being within 2% of the original methods on average across all datasets.

Concretely, we present our investigation of the following two questions. **Q1: Can we explain dimensions of a node embedding method with human-understandable graph features?** Examples of such

[1]Sometimes node embedding is referred to as graph embedding.

features are degree, clustering coefficient, eccentricity, and PageRank. These features can be any set of features that a practitioner decides to use and are not limited to graph features. Henderson et al. (2012) called such human-understandable features "sense" features. We will borrow their terminology in this paper. **Q2: Can we modify existing node embedding algorithms to produce embeddings that are explained with sense features? If so, how?** We present a new framework called XM (short for eXplain eMbedding) to answer this question. XM adds two constraints to an existing objective function for node embedding: (a) sparsity w.r.t. explaining an embedding dimension and (b) orthogonality between embedding dimensions. We provide an ablation study to analyze the impact of each constraint. XM outputs explainable node embeddings that are tested on a variety of real-world graphs, achieving downstream task performances comparable to the state of the art.

Our main contributions are as follows:

- We present a method for understanding what each dimension of a node embedding method means, with respect to a set of human-understandable sense features defined on nodes. This helps us to understand the placement of a node in the embedding space. These explanations are independent of any downstream task and are in the form of an *Explain* matrix, whose rows represent embedding dimensions and whose columns represent sense features.
- To evaluate Explain matrices, we use their nuclear norms and show that minimizing the nuclear norm of an Explain matrix removes noise and reduces the lower bound of its entropy, which is a desirable property.
- We introduce the XM framework, which modifies the objective function of any existing node embedding method to produce embeddings that provide better quality explanations. We analyze the impact of each of XM's constraints (sparsity and orthogonality) through an ablation study and show that using both constraints leads to the largest reduction in nuclear norms. We demonstrate the effectiveness of XM by modifying a number of embedding algorithms on different real-world datasets and evaluating them on the downstream task of link prediction.

2 Explaining embeddings

2.1 Sense-making

In order to explain each embedding dimension, we define a set of human-understandable node features, henceforth referred to as "sense" features. For this work, we use global and local properties of the network as our "sense" features. Following Ghasemian et al. (2020), we look at the following 15 features: degree, weighted degree (if the edges are weighted), clustering coefficient (Barabási, 2016), average of the personalized PageRank vector, standard deviation of the personalized PageRank vector, average degree of the neighboring nodes, average clustering coefficient of neighboring nodes, number of edges in the ego net, structural hole constraint (Burt, 2004), betweenness centrality (Freeman, 1977), eccentricity, PageRank, degree centrality, Katz centrality (Katz, 1953) and eigenvector centrality (Newman, 2010). Since some of these features are highly correlated, we choose to focus on the following subset of them throughout this work: degree, clustering coefficient, standard deviation of the personalized PageRank, average degree of the neighboring nodes, average clustering coefficient of neighboring nodes, eccentricity and Katz centrality.

Given an embedding vector for a node k, $\vec{y}_k \in \mathbb{R}^{d\times 1}$ and its corresponding sense feature vector $\vec{f}_k \in \mathbb{R}^{f\times 1}$ (where d is the number of dimensions of the embedding space and f is the number of sense features used, each being normalized between 0 and 1), we compute the element-wise similarity between the embedding vector for a node and its corresponding sense feature vector as follows:

$$E_k = \frac{\vec{y}_k \otimes \vec{f}_k^T}{\|\vec{y}_k\|\|\vec{f}_k\|} \tag{1}$$

We call $E \in \mathbb{R}^{d\times f}$ the *Explain* matrix (we drop the subscript k corresponding to node k, for brevity). Normalized to a range of 0 and 1, each row i of the Explain matrix corresponds to a dimension in the

embedding space and each column j corresponds to a sense feature. The value E_{ij} corresponds to how much the dimension i is defined by the sense feature j. This matrix E, helps us to investigate the linear relationships between the sense feature vectors and the embedding vector.

Structural vs Positional Sense Features

The set of features defined above is not exhaustive and was chosen for uniformity in various data sets. In cases where datasets have inherent node features, we ignore them and use the features defined above for a fair comparison across datasets. However, the choice of sense features is important, and we discuss the impact of different types of features in this section with toy examples.

Observe how the sense features defined above are all structural, meaning that they do not contain any information about the location or position of a node in the network. The sense feature vectors would be similar for two nodes with similar neighborhood structures in the same network. The explanations produced by these sense features would therefore be more role-based, i.e. would speak more to the role of a node in the network (e.g. gatekeeper node or hubs).

However, if the explanations one is looking for are more positional in nature,[2] structural sense features might not be very helpful. To demonstrate the differences in sense features, as well as what the Explain matrix contains, we use a simple barbell graph as a toy example (shown in Figure 1). We define a secondary set of positional sense features. We follow You et al. (2019) in defining the concept of anchor nodes and use simple features such as the number of hops to the anchor node and personalized PageRank of the anchor nodes as our positional sense features. We embed the graph using DGI and generate the Explain matrices for each node in the graph following Equation (1). Figures 1(A) and (B) show the Explain matrices for the bridge node in the top clique (pink) and a random node in the bottom clique (gray) using our simple positional features, while Figures 1(C) and (D) show the Explain matrices for the same nodes, but using the structural sense features described above. Observe in Figures 1(C) and (D) how degree and average neighbor clustering stand out for the bridge nodes (shown in pink) and features such as the clustering coefficient and average neighbor degree stand out for the other nodes (shown in gray). Note that in case of using structural sense features, all gray nodes (non-bridge nodes) would have similar sense features and embeddings; and therefore, similar Explain matrices, but, in the case of positional sense features (Figure 1(B)), the gray nodes in different cliques would have different sense feature vectors leading to different Explain matrices. (Observe that the feature "Hops To Anchor In Bottom Clique" stands out in Figure 1(B)).

Based on this toy example, it is clear that the set of 7 structural sense features used is not exhaustive. We use these 7 features for consistency across datasets but encourage researchers to tailor the chosen sense features to the desired type of explanations.

2.1.1 Real world example

We look at the Karate Club Network for a more concrete real-world example. We chose this network for brevity and pedagogy and discuss larger networks in the Results section. Figure 2(A) shows the original Karate Club Network (Zachary, 1977). We embed the Karate Club graph into 16 dimensions using DGI. We then visualize these embeddings by projecting them into 2 dimensions using UMAP (McInnes et al., 2018) (shown in Figure 2(B)). For a detailed node-wise view, we look at the president of the club (node 34), the instructor (node 1), and a random student (node 12). We use Equation (1) to compute the Explain matrix for each node. Figure 2(C) corresponds to the Explain matrix for the instructor (node 1), Figure 2(D) corresponds to the random student (node 12), and Figure 2(E) to the president (node 34). Examining the Explain matrices for nodes 1 and 34 (Figures 2(C) and (D)), we see similar sense features (like degree, the standard deviation of the personalized PageRank, Katz centrality and average neighbor clustering) stand out and are explained most by dimensions 14 and 15 (note that the specific dimensions are not constant). Also, observe that these two nodes are placed close together in the embedding space (low-dimensional visualization in Figure 2(B)). We also see that the Explain matrix for node 12 (Figure 2(D)) is different, and has average neighbor degree and eccentricity as its distinguishing features.

[2]Positional explanations look at the role of a node instead of its community.

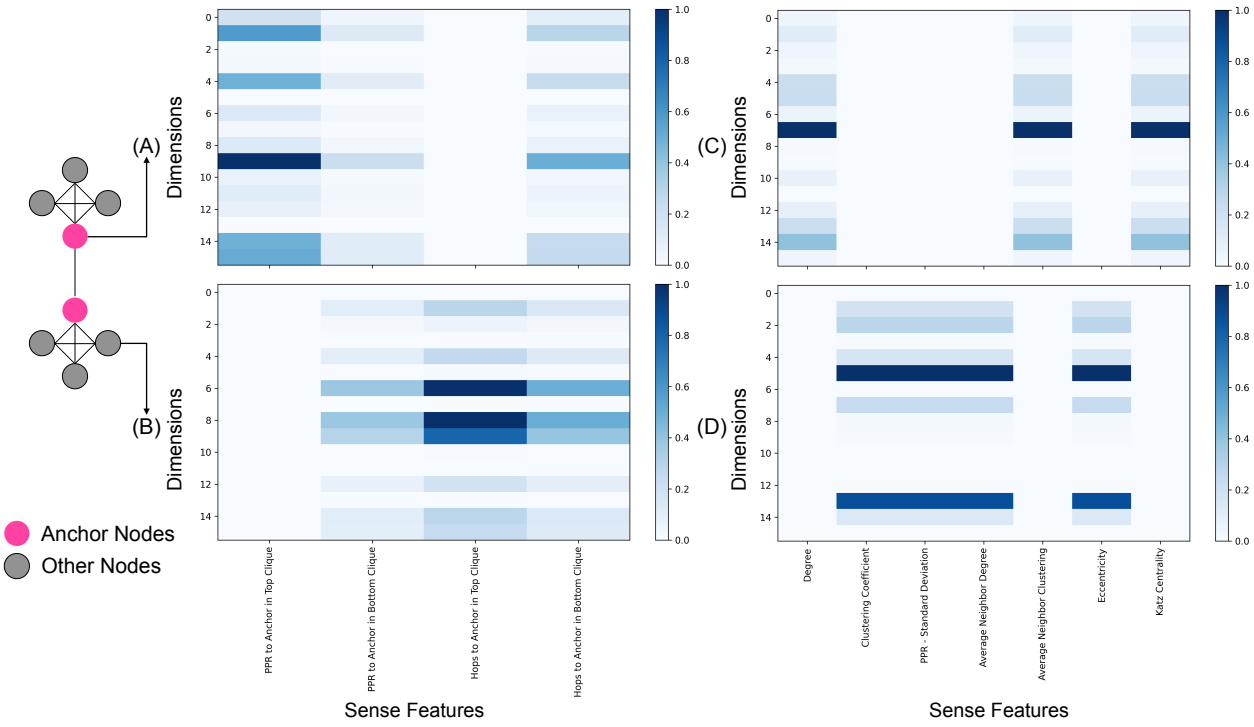


Figure 1: We show the sense making process using a simple toy example of a barbell graph. We embed the graph into 16 dimensions using DGI and show the Explain matrices for a bridge node (shown in pink) and a random node in a clique (shown in grey). The columns (x axis) of the Explain matrix correspond to the sense features and the rows (y axis) represent each embedding dimension. We demonstrate the utility of these Explain matrices and the differences seen when using different sets of sense features. (A)-(B) Explain matrices for the pointed nodes using positional sense features (shown on the x axis). (C)-(D) Explain matrices for the same two nodes, but using the structural sense features (shown on the x axis). Observe sub plot (C) where degree and average neighbor clustering stand out for the bridge nodes (green) and sub plot (D) where features like clustering coefficient and average neighbor degree stand out for the other nodes (grey). In the case of the structural sense features, all grey nodes would have similar Explain matrices given that they would all have similar sense features (i.e. degree, clustering coefficient etc..), but if passed in positional sense features instead, the grey nodes in different cliques would have different Explain matrices, as seen in sub plot (B) where the feature "Hops To Anchor In Bottom Clique" stands out.

2.2 Our Proposed Framework: XM (eXplain eMbedding)

Quantifying the Quality of Explanations

Given the method for generating explanations, we quantify the quality of these explanations. The (potentially) large number of dimensions and sense features drives us to use heatmaps to visualize their relationships. Figure 2 shows the heatmap visualization of the Explain matrix. Notice that a noisy Explain matrix is hard to interpret, but sparser Explain matrices with each dimension being defined by unique sets of sense features are easier to interpret.

Since what we are looking for is a denoised Explain matrix, we follow the work from image denoising using nuclear norm minimization [Gu et al. (2014), Xu et al. (2017)]. We report the nuclear norms of the Explain matrices as a quantitative metric, with lower nuclear norms leading to better explanations.

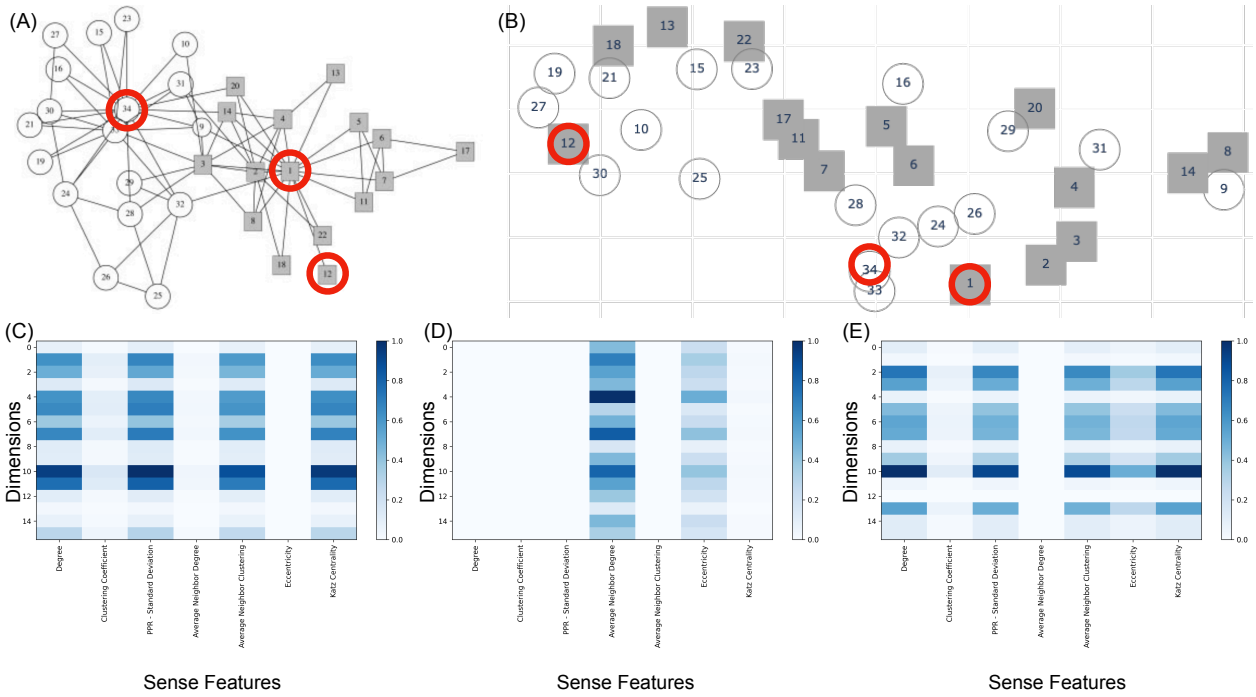


Figure 2: **(A)** Karate Club Network (Zachary, 1977). **(B)** We embed the Karate Club network into 16 dimensions using DGI and visualize it in 2 dimensions using UMAP (McInnes et al., 2018). Explain matrix for **(C)** the instructor (node 1). **(D)** Explain matrix for a random student node (node 12). **(E)** Explain matrix for the president (node 34). Examining the Explain matrices for each of these nodes gives us an understanding of their placement in the embeddings space. Observe how features such as degree, personalized page rank etc., stand out for nodes 1 and 34 (subplots **(C)** and **(E)**) and features like average neighbor degree stands out for the random student node (subplot **(D)**).

2.2.1 Augmenting Embeddings

Note that we do not want a perfect one-to-one mapping between the embedding dimensions and the sense features. If a one-to-one correspondence is imposed, then the network information beyond that contained in the simple node sense features is lost while creating the embeddings. Such embeddings would not capture higher-order network properties that contribute to downstream tasks (Ghasemian et al., 2020), resulting in poor performance. Using sense features themselves as embedding vectors or performing dimensionality reduction on a large set of sense features suffers from the same issue of not capturing higher-order network properties. To that end, we propose XM, a framework for augmenting existing embedding algorithms that includes sense feature information, as well as information that the underlying algorithm captures.

XM incorporates sense features and achieves a less noisy Explain matrix by imposing a sparsity constraint along the columns of the Explain matrix. This aims to zero-out features with small contributions to the definition of a dimension. Following recent advances in dimensional contrastive learning by Nguyen et al. (2023), XM also imposes an orthogonality constraint along the rows of the Explain matrix which aims to have embedding dimensions that are defined by different sets of sense features. XM uses the two constraints, sparsity and orthogonality, instead of directly minimizing the nuclear norm for added control over the objective function (see section 3.4). The loss terms for the two constraints can be written as follows.

$$L_{sparse} = \|E_k[:, j]\|_1 \qquad \forall j \in columns \tag{2}$$

$$L_{ortho} = \|E_k[i, :]E_k^T[j, :]\|_2 \qquad \forall i \in rows \; \forall j \in columns \tag{3}$$

Table 1: Embedding algorithms used. Classification of these algorithms was adopted from Chami et al. Chami et al. (2022), where we pick two algorithms that allow for a node feature matrix, and two algorithms that do not. Providing sense features as node attributes is a simple method of incorporating the sense features into the embeddings. XM variants outperform the original versions in terms of nuclear norms.

Algorithm	Node Features	Sub Type
SDNE Wang et al. (2016)	No ($X = I$)	Autoencoder
LINE Tang et al. (2015)	No ($X = I$)	Skip Gram
DGI Velickovic et al. (2019)	Yes ($X \neq I$)	Message Passing
GMI Peng et al. (2020)	Yes ($X \neq I$)	Message Passing

Table 2: Real-world networks used in our experiments. $\langle k \rangle$ refers to the average degree, σ_k refers to the standard deviation of the degree, r refers to the degree assortativity and c refers to the average clustering coefficient. These networks were picked to span a range of average degrees and clustering coefficients.

Network	Nodes	Edges	$\langle k \rangle$	σ_k	r	c	Transitivity
EU Email Leskovec et al. (2007)	986	16687	33.84	37.81	-0.01	0.40	0.26
US Airport Zhu et al. (2021)	1186	13597	22.92	40.49	0.03	0.50	0.42
Squirrel Rozemberczki et al. (2021)	5201	198493	76.32	161.45	-0.23	0.42	0.34
Citeseer Bollacker et al. (1998)	3327	4676	2.81	3.38	0.05	0.14	0.13
FB15K-237 Toutanova & Chen (2015)	14951	261581	34.99	111.43	-0.10	0.213	0.02
PubMed Roberts (2001)	19717	44327	4.49	7.43	-0.04	0.06	0.05

These two additional loss terms are then added to the objective functions of existing embedding algorithms. As an example, the objective for SDNE consists of 3 loss terms - a first order loss term, a reconstruction error term and a regularization term:

$$L_{SDNE} = \alpha L_{first} + \beta L_{recon} + \nu L_{reg} \tag{4}$$

where α, β, and ν are tunable hyperparameters.

We augment this objective as follows :

$$L_{SDNE} = \alpha L_1 + \beta L_2 + \nu L_{reg} + \gamma L_{Sparse} + \delta L_{Ortho} \tag{5}$$

which we refer to as SDNE+XM. Here, γ and δ are hyperparameters for the sparsity and orthogonality constraints, respectively.

In order to demonstrate our method, we apply our augmentation to 4 unsupervised embedding algorithms across 6 different networks. We look at the taxonomy of machine learning on graphs by Chami et al. (2022) and pick two algorithms from the branch where $X = I$, (i.e. these algorithms do not incorporate node attributes) and two algorithms from the branch where $X \neq I$ (i.e. these algorithms do allow node attributes). We choose these algorithms because providing sense features as node attributes is a simple and straightforward method to incorporate the sense features into the embeddings. However, as we show in the Results section, we achieve lower nuclear norms by augmenting the objective function.

Table 1 shows the algorithms selected and Table 2 summarizes the datasets used. These have been chosen to span a range of average degrees and clustering coefficients.

Interpreting Rank Reduction with Nuclear Norms

Given a matrix of rank r, one can reconstruct it using r rank-1 matrices. The nuclear norm of a matrix is the tightest convex bound of the rank function over the unit ball of matrices with bounded spectral norm.[3]

[3]Note that the rank of a matrix is a non-convex function, which makes optimization problems involving the rank hard to solve. The nuclear norm serves as a convex proxy for the rank.

Thus, the difference in nuclear norms of two matrices gives us a lower bound on how many more rank-1 matrices are needed to reconstruct the matrices. For example, SDNE on the EU Email dataset (with $d = 128$) produces an Explain matrix whose nuclear norm is $\simeq 8 \pm 0.97$ (see Figure 7). However, its Explain matrix has a nuclear norm of $\simeq 1.8 \pm 0.26$ when the sparsity constraint is used, $\simeq 2.1 \pm 0.29$ when the orthogonality constraint is used, and $\simeq 1.6 \pm 0.39$ when both constraints are used. (Note that the error bars throughout denote standard error) This means that the Explain matrix for the EU Email dataset can be approximated with two rank-1 matrices when the sparsity and orthogonality constraints are used, instead of eight rank-1 matrices with no constraints used.

Nuclear Norm and Matrix Entropy

For a given node, assume there exists an embedding y^* (we drop the vector notation $\vec{y}$ for brevity), which is the optimal embedding. Let y^{xm} be the embedding for the same node found by XM. Recall that we define the Explain matrix E as:

$$E^* = \frac{y^* \otimes f^T}{\|y^*\|\|f\|} \tag{6}$$

$$E^{xm} = \frac{y^{xm} \otimes f^T}{\|y^{xm}\|\|f\|} \tag{7}$$

Let

$$A = E^* {E^*}^T \tag{8}$$

$$B = E^{xm} {E^{xm}}^T \tag{9}$$

Observe that A and B are now symmetric matrices and therefore have an orthonormal basis of eigenvectors with real eigenvalues.

We then have

$$A = U\Sigma U^T \qquad \text{where } U\text{is orthonormal and } \Sigma \text{ is diagonal} \tag{10}$$

$$z^T A z = z^T \; U\Sigma U^T \; z \qquad \forall z \geq 0 \in \mathbb{R}^n \tag{11}$$

$$z^T A z = \sum_{i=1}^{n} \Sigma_{ii} (U^T z)_i^2 \tag{12}$$

$$z^T A z \geq 0 \tag{13}$$

This shows that A is a positive semi-definite (PSD) matrix. The same proof applies for the matrix B. Given that A and B are PSD, Yu (2013) and Bach (2022) show that when the Von Neumann entropy of a PSD matrix is defined as

$$H(A) = -tr[A \; log \; A] \tag{14}$$

and its associated Bregman Divergence is defined as

$$D(A\|B) = tr[A(log \; A - log \; B)] - tr(A) + tr(B) \tag{15}$$

the following extension of Pinsker's inequality holds true:

$$D(A\|B) \geq \frac{1}{2}\|A - B\|_*^2 \tag{16}$$

We also know that the nuclear norm is an upper bound on the L_2 norm, i.e.

$$\|A\|_* \geq \|A\|_2 \tag{17}$$

This gives us

$$D(A\|B) \geq \frac{1}{2}\|A - B\|_*^2 \geq \frac{1}{2}\|A - B\|_2^2 \tag{18}$$

$$D(A\|B) \geq \frac{1}{2}[\|A\|_2^2 - 2\|A\|_2\|B\|_2 + \|B\|_2^2] \tag{19}$$

Recall that

$$B = E^{xm}E^{xm^T} \tag{20}$$

which is explicitly minimized as the orthogonality constraint in the XM framework. Notice that the $\|A\|_2$ term is constant since it is the optimal solution. Thus, by minimizing the L_2 norm of B, we reduce the lower bound in the entropy between an optimal explanation E^* and the explanation E^{xm} found by XM.

3 Experiments

We start by visualizing the Explain matrices for the Karate Club network embedded using DGI. We then discuss quantitative metrics for the networks from Table 2 embedded using each of the 4 embedding algorithms from Table 1 and their XM variants.

3.1 Explanations - Visual Evaluation

Figure 3, shows the Explain matrix for the Karate Club network. As in Figure 2 (C)(D) and (E), the three subplots of Figure 3 show the Explain matrices for the instructor node (node 1, Fig 3(A)), a random student node (node 12, Fig 3(B)) and the president node (node 34, Fig 3(C)). To stay consistent across experiments, these plots were generated using 16-dimensional embeddings that were generated from DGI+XM. Observe that the Explain matrices shown in Figure 3(A)(B) and (C), while highlighting similar sense features, are sparser than the respective plots in Figure 2(C)(D) and (E) and each sense feature is defined by fewer dimensions, i.e. the dimensions are orthogonal in their explanations when compared to the standard DGI.

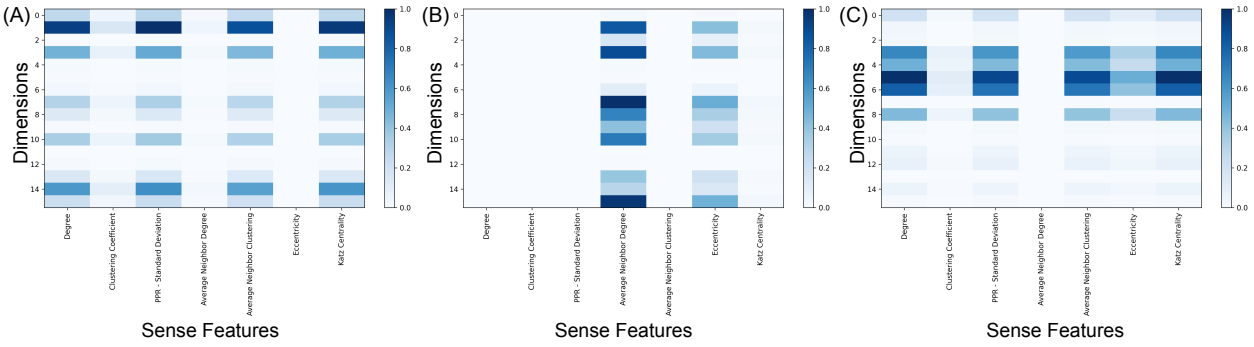


Figure 3: Explain matrices for the Karate Club network (Zachary, 1977) generated using DGI+XM for (A) the instructor (node 1), (B) a random student (node 12), and (C) the president (node 34). Compare each of (A), (B), and (C) to Figure 2(C),(D), and (E), respectively. Observe that the Explain matrices in this figure (generated by DGI+XM) are sparser and each sense feature is explained by fewer dimensions, when compared to the Explain matrices from Figure 2 (generated by the standard DGI).

3.2 Explanations - Quantitative Evaluation

We look at the nuclear norms of the Explain matrices as a quantitative metric to evaluate and compare the standard and XM variants of the algorithms. For each dataset, we use each of the 4 algorithms shown in Table 1 and compute the mean of the nuclear norm of the Explain matrices across all nodes in the dataset. We embed every dataset into 128 dimensions for consistency across datasets and algorithms and pass in sense features as node attributes when running DGI and GMI. Figure 4 shows the distribution of the nuclear norms of the Explain matrices for SDNE and SDNE+XM with results for the other algorithms shown in the appendix (Figures 8, 9, and 10). We see that the distribution of the nuclear norm values shift to the left (i.e. lower) across each dataset.

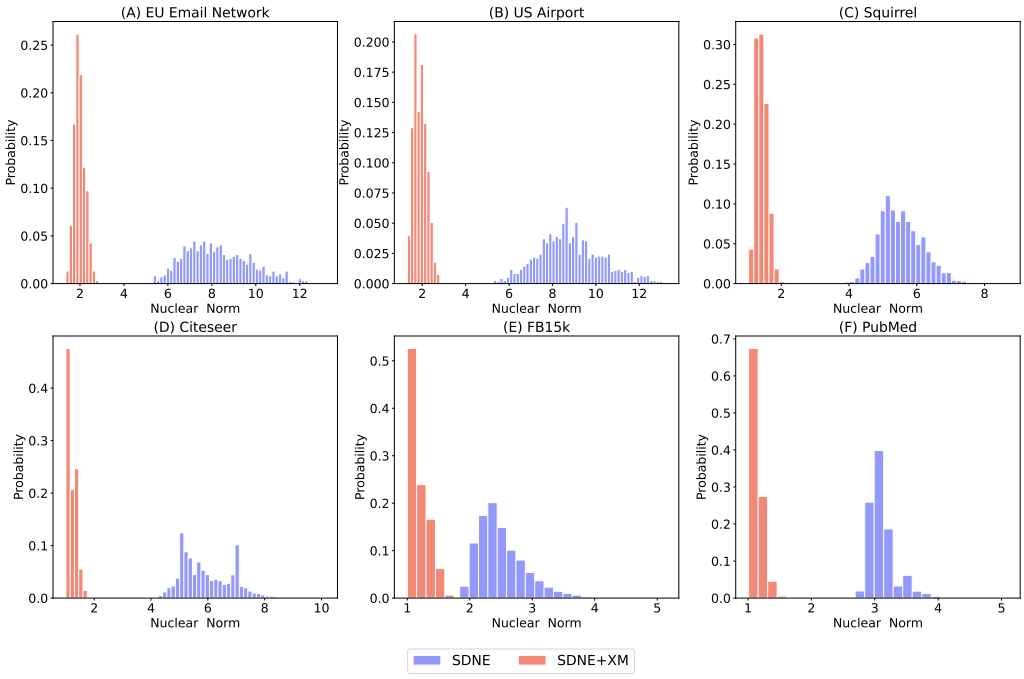


Figure 4: Distribution of nuclear norms of the Explain matrices for each node for SDNE and SDNE+XM. Observe how the distribution of SDNE+XM is shifted to the left with a lower mean across each of the 6 datasets. See Section A.2 in the appendix for results on the other methods.

3.3 Link Prediction Performance

By adding additional loss terms to the objective function, we modify the embeddings themselves to generate a less noisy Explain matrix. Doing so seems to have an impact on downstream tasks, although, we still see comparable performance to the original version. We use link prediction as our downstream task and sample an equal number of positive and negative edges from the graph. We use 60% of the edges as training data and the remaining as test data. The embeddings are passed through a simple 2 layer fully connected neural network. We repeat the entire process 3 times to create a three-fold cross validation setup. We use 128-dimensional embeddings across all algorithms and networks for consistency.

Results are shown in Figure 5 with nuclear norm shown on the $y-$axis and the AUC shown on the $x-$axis with error bars denoting the standard error. We observe that across all algorithms, AUC scores are comparable to the original version with the corresponding nuclear norms being lower.

We also examine differences in runtime due to the additional constraints in Figure 6. More specifically, we examine the time (in seconds) per epoch for each of the 4 algorithms we test for the EU Email network. We observe that the XM variants are comparable in runtime to their original versions. Results shown are averaged across 5 runs with error bars showing standard error. Similar run time comparisons for other datasets are shown in the appendix (Figures 11 through 15).

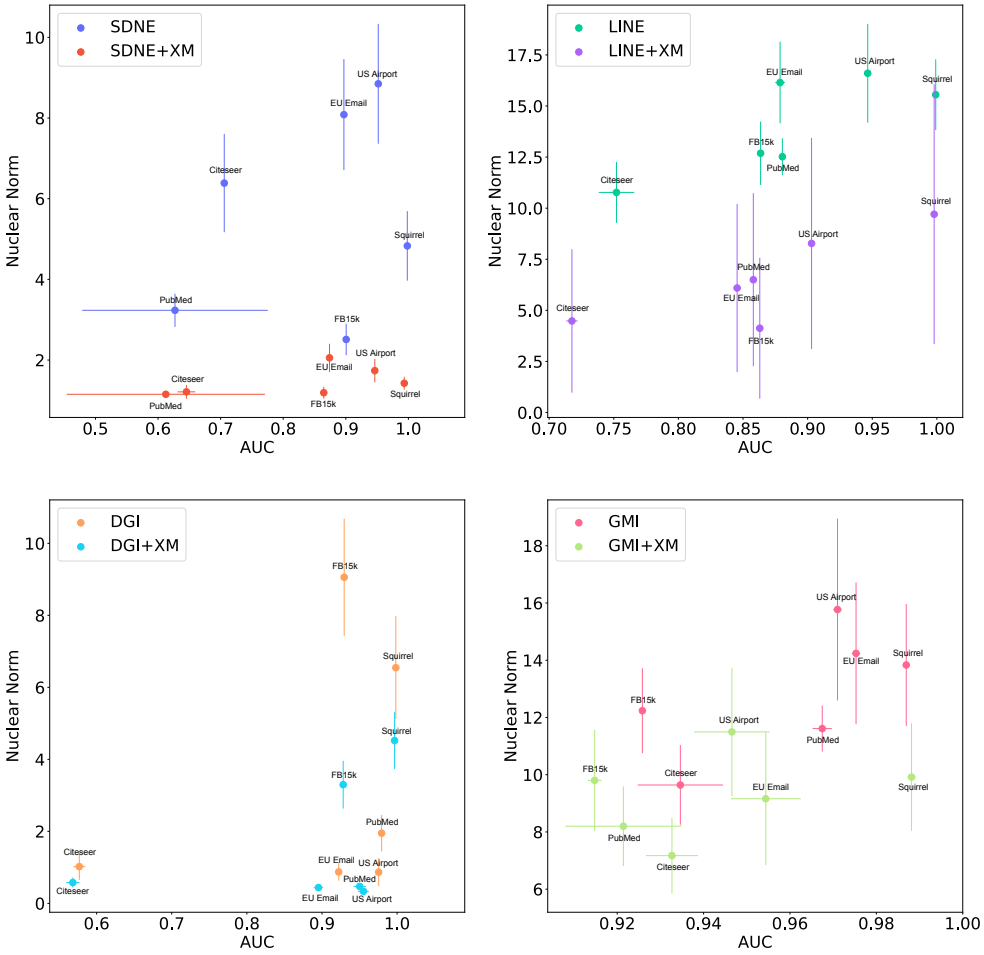


Figure 5: AUC scores for link prediction across 4 embedding algorithms - SDNE, LINE, DGI, GMI - shown in each subplot, on the 6 networks defined in Table 2. Error bars show standard error. AUC scores for the XM variants are comparable to the original algorithms with nuclear norms of the XM variants being lower. The XM variants also follow the original algorithms in terms of standard error, for example, SDNE has large variance in AUC for the PubMed dataset, which can also be seen for SDNE+XM. Results are from a three-fold cross validation experiment with 128 dimensional embeddings.

3.4 Ablation Study

We add two constraints, orthogonality and sparsity, while computing the augmented node embeddings. We study how each contributes to the reduction of nuclear norms. Figure 7 shows nuclear norms for each network embedded in 128 dimensions using each of the 4 embeddings algorithms, averaged across 5 runs. We see that the orthogonality constraint alone reduces the nuclear norm more than the sparsity constraint alone. We see that across all algorithms, using both constraints together achieves the lowest nuclear norm.

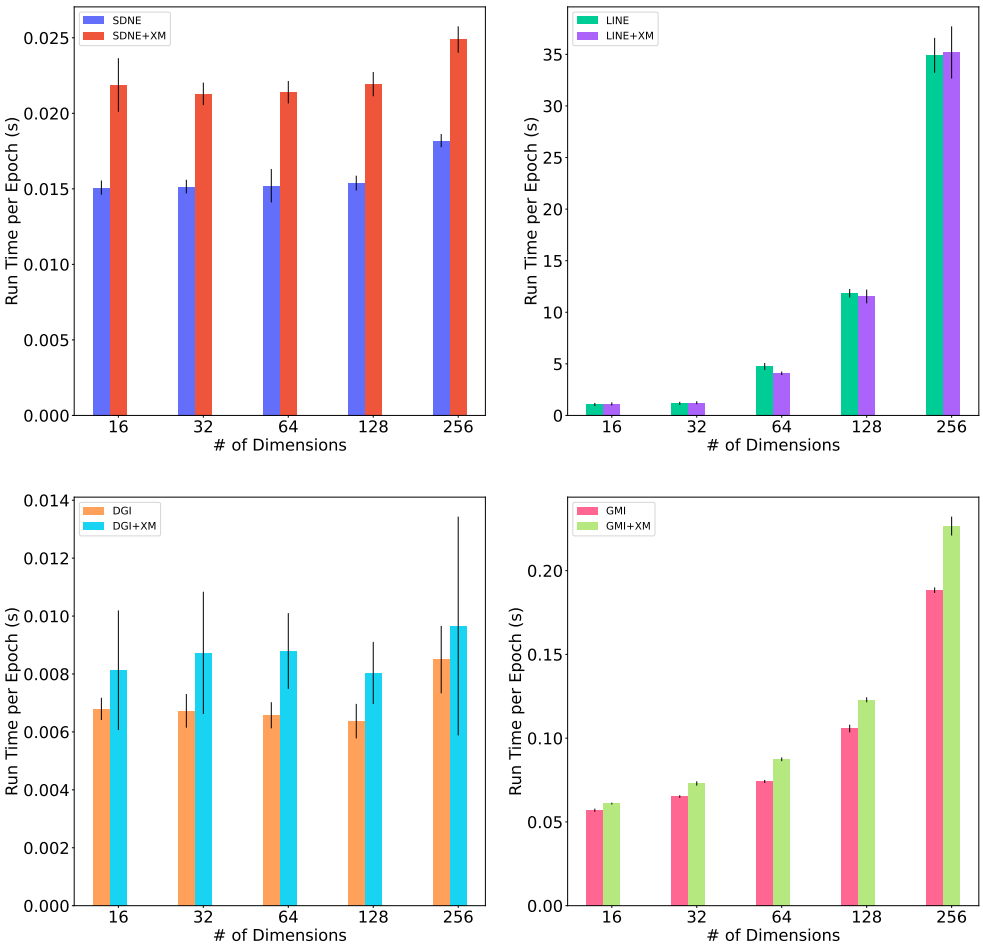


Figure 6: EU email network: runtimes per epoch (in seconds) for SDNE, LINE, DGI, GMI, and their corresponding XM variants. Results are averaged across 5 runs. Observe that the runtimes per epoch are comparable to the original versions. Error bars show standard error. See Section A.3 in the Appendix for the resutls on other networks.

4 Discussion

By design, the explanations are granular. They are for each node at the granularity of each dimension. This is useful when inspecting particular nodes, but to get a holistic explanation of a large network will require further analysis of a set of Explain matrices.

As is common in works on explainability, the use of the XM framework leads to a marginal reduction in performance scores for downstream tasks. However, note that the reduction in performance can be controlled through the choice of hyperparameters. For the this study, we chose to prioritize the quantitative value of the nuclear norm to get the largest reduction possible. In actual use, the practitioner can balance the amount of denoising of the Explain matrices to the desired level of performance in downstream tasks by setting the hyperparameters accordingly. (Our choices for the hyperparameters are available in our online code repository at https://anonymous.4open.science/r/ExplainingNodeEmbeddings-FE6C/.)

The usability of XM depends on the choice of sense features provided. A poor choice could lead to Explain matrices that do not contain any useful information. The decision to use structural sense features throughout this work was to provide a set of features that are reasonable and understandable to humans in the graph

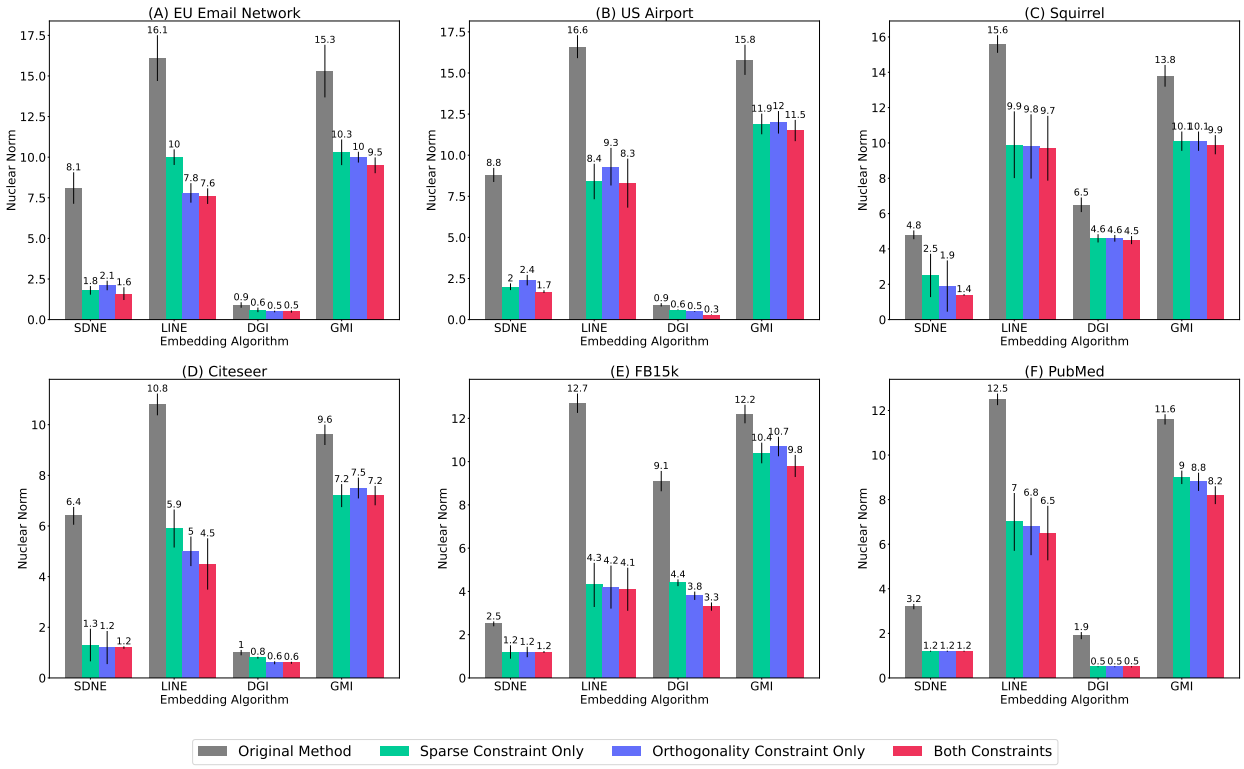


Figure 7: Ablation study examining the impact of each term in the loss function. Observe that the orthogonality constraint alone reduces nuclear norm more than the sparsity constraint alone. However, using both constraints leads to the lowest nuclear norms. The plot shows average values over 5 runs. Results shown are for 128 dimensional embeddings but hold true across various dimensions. (We tried $d = 16, 32, 64, 128, 256$.) We do not find a correlation between the number of embedding dimensions and the reduction in the nuclear norm values. Error bars show standard error.

context (e.g. node degree and clustering coefficient) and can be used in networks from different domains (e.g. sociology and biology).

5 Related Work

Interest in explainable machine learning models has attracted much attention in recent years, in part because the performance of such models can often be misleading (Burkart & Huber, 2021). With the increasing application of machine learning to graphs, the development of explainable models is becoming a necessity. However, most of the work on explainable models focuses on explaining predictions for downstream tasks. For example, Pope et al. (2019) and Baldassarre & Azizpour (2019) extend the concept of explainable predictions in images to network data using gradient-based approaches to visualize and understand the predictions of Graph Convolutional Networks (GCNs).

Another popular set of methods focuses on finding a subgraph that contributes to a particular classification. For example, Ying et al. (2019) introduce *GNNExplainer*, which attempts to find a subgraph and a subset of the node features by maximizing the mutual information between the original prediction and the prediction about a substructure. However, the explanations generated are customized for each instance, which makes them hard to generate. Luo et al. (2020) extend *GNNExplainer* by introducing *PGExplainer*, which utilizes a generative probabilistic model for graph data that has been shown to be able to learn underlying structures from observed data. It uses these underlying structures as explanations and models the underlying structure as edge distributions, where the explanatory graph is sampled. This generative model in *PGExplainer*

is parameterized with a deep neural network to collectively explain the predictions of multiple instances. Spinelli et al. (2022) propose a meta-explainer to improve the quality of explanations during training time. Roughly related to our approach, they aim to steer the optimization procedure towards minima that allow post-hoc explainers like *GNNExplainer* and *PGExplainer* to achieve better results. Vu & Thai (2020) use probabilistic graphical models to explain substructures in a network, and Yuan et al. (2021) use Monte Carlo tree search to search all possible subgraphs and identify important substructures.

Zhang et al. (2021b) propose RelEx, which identifies important nodes and links in the prediction of a given node, but does so by treating the underlying model as a black box and learning relational explanations. Explanations come in the form of a mask matrix, which, akin to our work, has sparsity constraints. Along the lines of sparsity, Lin et al. (2020) propose GISST (Graph neural networks Including SparSe inTerpretability), which combines an attention mechanism and sparsity regularization to yield an important subgraph and node feature subset related to any graph-based task. Li et al. (2022b) use a student-teacher setup to distill knowledge from a large pre-trained GNN to a shallower GNN, while explicitly learning contribution weights between two nodes using an attention mechanism. Schnake et al. (2021) propose GNN-LRP that uses layer-wise relevance propagation that outputs a collection of walks that are relevant for a given prediction.

Agarwal et al. (2023) provide a framework titled GraphXAI that delves deeper into the different types of GNN explainers. These explainers can be categorized into three types: perturbation-based, gradient-based, and surrogate-based models. These explainers can often be misleading, as the explanations may rely on a different rationale for the test data compared to the rationale learned during training. GraphXAI provides a benchmark for evaluating these explainers by generating ground-truth explanations using the ShapeGGen graph generator, which is capable of generating ground-truth explanations for homophilic, heterophilic, and attributed graphs. We refer the interested reader to the survey by Li et al. (2022a), which provides a detailed taxonomy of the various graph explanation methods.

All of these explanatory methods depend on a downstream task such as link prediction or node classification. More closely related to our work (i.e. explanations independent of a downstream task), are works by Dalmia et al. (2018) and Bonner et al. (2019) who study how good an embedding algorithm is at explaining certain node properties (akin to what we have referred to as sense features). Liu et al. (2018) utilize network homophily and hierarchically cluster nodes based on the embeddings to create a taxonomy. Explanations come in the form of these taxonomies. Also related would be the study of word analogies in the natural language domain where one could interpret a word as a node in a graph. Ethayarajh et al. (2018) and Allen & Hospedales (2019) explore the relationships between word embeddings and analogies, however, these are applicable only to certain categories of word embeddings (negative sampling-based methods). Yin & Shen (2018) provide a theoretical understanding of word embeddings and its dimensionality and propose a metric on dissimilarity between word embeddings. However, they focus on embedding algorithms that can be formulated as matrix factorizations.Monti et al. (2024) provide an axiomatic approach for auditing explainers for the task of node classification. More specifically, they define a set of 'important' features such that a change in these 'important' features would cause a change in the classification result and check if an explainer is able to diagnose if an underlying model is using the 'important' feature or not.

In terms of explanations per dimension, Gogoglou et al. (2019) define an interpretability score for each dimension in an node embedding. While they are similar in that they also provide explanations for each dimension, the explanations are in the form of an interpretability score that measures how well a dimension is associated with a subgroup of nodes. Khoshraftar et al. (2021) define the interpretability of an embedding algorithm as the extent to which its embedding dimensions can represent "important" nodes in a category with extreme values, where "importance" is based on centrality measures. Carrying over work from traditional explanation methods like SHAP and LIME, Gui et al. (2022) propose FlowX, a method of explaining GNNs by identifying important message flows using Shapley values. A flow here is defined as the set of T messages that are passed in a message passing network with T layers. Each flow for a given node is then treated as a player for the Shapley formulation. Huang et al. (2022) propose GraphLIME that learns an interpretable model locally in the subgraph of the node being explained. As discussed earlier, Yuan et al. (2021) use Monte Carlo tree search to explore subgraphs, but use Shapley values to compute subgraph importance. Duval & Malliaros (2021) propose GraphSVX that captures the contribution of each feature and node towards the

prediction by constructing a surrogate model on a perturbed dataset. Here, the explanations come in the form of Shapley values.

In terms of objective functions, Duong et al. (2019) propose a loss function that uses a notion of minimizing the number of connections between communities, and in doing so, the nodes in a community are embedded closer to each other. Explainability comes in the form of each dimension defining the degree of membership to a community. Rossi et al. (2018) use features akin to what we refer to as sense features as base features, to which relational feature operators are applied to generate embeddings. Explainability comes in the form of examining which relational operators were used to generate the embeddings.

Compared with related work in the field, our proposed method XM is independent of any particular downstream task, can be used to extend any existing embedding algorithm, and works with any set of sense features. It also provides explanations on the node- and dimension-level.

6 Conclusion and Future Work

We present a method for generating explanations for each node in the form of an Explain matrix using a set of human-understandable "sense" features. These features can be based on the graph structure (such as degree, clustering coefficient, and PageRank) or other user-defined features. We use the nuclear norm of the generated Explain matrices to quantify their usefulness. The choice of nuclear norm is due to its relationship with the entropy of a matrix. In particular, reducing the nuclear norm leads to the reduction of the lower bound of the entropy of the Explain matrices. Moreover, we introduce the XM framework, which modifies existing embedding algorithms to produce embeddings with low nuclear norms on Explain matrices, while maintaining downstream task performance. We demonstrate the effectiveness of XM across 4 embedding algorithms and 6 real-world datasets and analyze the impact of XM's constraints through an ablation study. XM is independent of any downstream task and can be used with any existing embedding algorithm and any set of sense features.

Future work. We noticed that the Explain matrices tend to have dimensions that do not contribute to explaining any sense features – i.e. the rows corresponding to these dimensions in the Explain matrix are near zero. Such information can be used for model selection – i.e. to set the hyperparameter associated with the number of embedding dimensions.

References

Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. Evaluating explainability for graph neural networks. *Scientific Data*, 2023.

Carl Allen and Timothy Hospedales. Analogies explained: Towards understanding word embeddings. In *ICML*, 2019.

Francis Bach. Playing with positive definite matrices – ii: entropy edition. `https://francisbach.com/von-neumann-entropy/`, 2022.

Federico Baldassarre and Hossein Azizpour. Explainability techniques for graph convolutional networks. *arXiv preprint arXiv:1905.13686*, 2019.

Albert-László Barabási. *Network Science*. Cambridge University Press, 2016. ISBN 9781107076266.

Kurt D. Bollacker, Steve Lawrence, and C. Lee Giles. Citeseer: An autonomous web agent for automatic retrieval and identification of interesting publications. In *International Conference on Autonomous Agents*, 1998.

Stephen Bonner, Ibad Kureshi, John Brennan, Georgios Theodoropoulos, Andrew Stephen McGough, and Boguslaw Obara. Exploring the semantic content of unsupervised graph embeddings: An empirical study. *Data Science and Engineering*, 4:269–289, 2019.

Nadia Burkart and Marco F Huber. A survey on the explainability of supervised machine learning. *JAIR*, 70:245–317, 2021.

Ronald S Burt. Structural holes and good ideas. *Am. J. Sociol.*, 110(2):349–399, 2004.

Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy. Machine learning on graphs: A model and comprehensive taxonomy. *JMLR*, 23(89):1–64, 2022.

Ayushi Dalmia, Ganesh J, and Manish Gupta. Towards interpretation of node embeddings. In *WWW*, pp. 945–952, 2018.

Chi Thang Duong, Quoc Viet Hung Nguyen, and Karl Aberer. Interpretable node embeddings with mincut loss. In *ICML (Workshop)*, 2019.

Alexandre Duval and Fragkiskos D Malliaros. Graphsvx: Shapley value explanations for graph neural networks. In *ECML PKDD*, pp. 302–318, 2021.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. Towards understanding linear word analogies. *arXiv preprint arXiv:1810.04882*, 2018.

Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

Amir Ghasemian, Homa Hosseinmardi, Aram Galstyan, Edoardo M. Airoldi, and Aaron Clauset. Stacking models for nearly optimal link prediction in complex networks. *PNAS*, 117(38), 2020.

Antonia Gogoglou, C Bayan Bruss, and Keegan E Hines. On the interpretability and evaluation of graph representation learning. *arXiv preprint arXiv:1910.03081*, 2019.

Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *IEEE CVPR*, pp. 2862–2869, 2014.

Shurui Gui, Hao Yuan, Jie Wang, Qicheng Lao, Kang Li, and Shuiwang Ji. Flowx: Towards explainable graph neural networks via message flows. *arXiv preprint arXiv:2206.12987*, 2022.

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 *ACM SIGKDD*, 2012.

Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang. Graphlime: Local interpretable model explanations for graph neural networks. *IEEE TKDE*, 2022.

Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

Shima Khoshraftar, Sedigheh Mahdavi, and Aijun An. Centrality-based interpretability measures for graph embeddings. In *IEEE DSAA*, pp. 1–10, 2021.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM TKDD*, 2007.

Yiqiao Li, Jianlong Zhou, Sunny Verma, and Fang Chen. A survey of explainable graph neural networks: Taxonomy and evaluation metrics. *arXiv preprint arXiv:2207.12599*, 2022a.

Yuan Li, Li Liu, Guoyin Wang, Yong Du, and Penggang Chen. Egnn: Constructing explainable graph neural networks via knowledge distillation. *Knowledge-Based Systems*, 241:108345, 2022b.

Chris Lin, Gerald J Sun, Krishna C Bulusu, Jonathan R Dry, and Marylens Hernandez. Graph neural networks including sparse interpretability. *arXiv preprint arXiv:2007.00119*, 2020.

Ninghao Liu, Xiao Huang, Jundong Li, and Xia Hu. On interpretation of network embedding via taxonomy induction. In *ACM SIGKDD*, pp. 1812–1820, 2018.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *NeurIPS*, 33:19620–19631, 2020.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018.

Nicolas Menand and C Seshadhri. Link prediction using low-dimensional node embeddings: The measurement problem. *PNAS*, 121(8), 2024.

Corrado Monti, Paolo Bajardi, Francesco Bonchi, André Panisson, and Alan Perotti. A true-to-the-model axiomatic benchmark for graph-based explainers. *TMLR*, 2024.

M. Newman. *Networks: An Introduction*. OUP Oxford, 2010.

Thanh Nguyen, Trung Xuan Pham, Chaoning Zhang, Tung M Luu, Thang Vu, and Chang D Yoo. Dimcl: Dimensional contrastive learning for improving self-supervised learning. *IEEE Access*, 11:21534–21545, 2023.

Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. Graph representation learning via graphical mutual information maximization. In *WWW*, pp. 259–270, 2020.

Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *IEEE CVPR*, pp. 10764–10773, 2019.

Richard J Roberts. Pubmed central: The genbank of the published literature, 2001.

Ryan A Rossi, Rong Zhou, and Nesreen K Ahmed. Deep inductive graph representation learning. *IEEE TKDE*, 32(3):438–452, 2018.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2), 2021.

Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T Schütt, Klaus-Robert Müller, and Grégoire Montavon. Higher-order explanations of graph neural networks via relevant walks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7581–7596, 2021.

Indro Spinelli, Simone Scardapane, and Aurelio Uncini. A meta-learning approach for training explainable graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Andrew Stolman, Caleb Levy, C Seshadhri, and Aneesh Sharma. Classic graph structural features outperform factorization-based graph embedding methods on community labeling. In *SDM*, pp. 388–396, 2022.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *WWW*, pp. 1067–1077, 2015.

Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pp. 57–66, 2015.

Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *ICLR (Poster)*, 2(3):4, 2019.

Minh Vu and My T Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *NeurIPS*, 33:12225–12235, 2020.

Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *ACM SIGKDD*, 2016.

Jun Xu, Lei Zhang, David Zhang, and Xiangchu Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *IEEE ICCV*, pp. 1096–1104, 2017.

Zi Yin and Yuanyuan Shen. On the dimensionality of word embedding. *NeurIPS*, 31, 2018.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *NeurIPS*, 32:1–12, 2019.

Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *ICML*, pp. 7134–7143, 2019.

Yao-Liang Yu. The strong convexity of von neumann's entropy. *Unpublished note*, 2013.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *ICML*, volume 139, pp. 12241–12252, 2021.

Wayne W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452—-473, 1977.

Yi-Jiao Zhang, Kai-Cheng Yang, and Filippo Radicchi. Systematic comparison of graph embedding methods in practical tasks. *Phys. Rev. E*, 104(4):044315, 2021a.

Yue Zhang, David Defazio, and Arti Ramesh. Relex: A model-agnostic relational model explainer. In *AIES*, pp. 1042–1049, 2021b.

Jing Zhu, Xingyu Lu, Mark Heimann, and Danai Koutra. *Node Proximity Is All You Need: Unified Structural and Positional Node and Graph Embedding*, pp. 163–171. 2021.

A Appendix

A.1 Variations to the Objective Function

Recall that we define the Explain matrix for a node k:

$$E_k = \frac{\vec{y_k} \otimes \vec{f_k}^T}{\|\vec{y_k}\|\|\vec{f_k}\|} \tag{21}$$

Thus, the sparsity and orthogonality constraints (as defined in Equations (2) and (3), respectively) are on nodes. To add pairwise constraints on nodes, we investigate variations to Equation (3).

Specifically, we implement the following variations for any node pair i and j :

1. Minimize $a_{ij}\|E_i - E_j\|_F$.
   This constrains the E matrices of each pair of nodes i, j to be similar if they are connected by an edge (i.e. if $a_{ij} = 1$), where the similarity is captured in terms of the Frobenius norm of the difference between their two Explain matrices.

2. Minimize $\|E_i E_j^T - f_i f_j^T\|_F$.
   Here, we move away from asking whether or not two nodes are connected by an edge. We compute the dimension-wise similarities between the Explain matrices and the similarities between sense feature vectors for all possible node pairs and obtain two distance matrices, each of dimension $\mathbb{R}^{n\times n}$. We then minimize the Frobenius norm of the difference between these two matrices.

We find that adding pairwise constraints to the objective function is not a good idea. The outcome is very sensitive to the choice of hyperparameters and the reduction in nuclear norm is not as significant. We posit that this may be due to opposing components of the objective function. For example, consider a high-degree node (a.k.a. a hub) that is connected to a low-degree node. In this case, most algorithms place the embeddings of the two nodes close to each other (since the nodes are connected) even though their sense features are different. To summarize, we find that using the formulation defined in Equations (2) and (3) leads to the most stable setup with the largest reduction in nuclear norms among the ones described above.

A.2 Nuclear Norm Distributions for Explain Matrices

Recall that each node has a corresponding Explain matrix. Thus, for each algorithm and dataset, we calculate the nuclear norm distribution across the Explain matrices of nodes. Figures 8, 9, and 10, respectively, show the nuclear norm distributions between LINE and LINE+XM, DGI and DGI+XM, and GMI and GMI+XM for all 6 datasets. We observe that the XM variants produce Explain matrices whose nuclear norms are lower than the original versions.

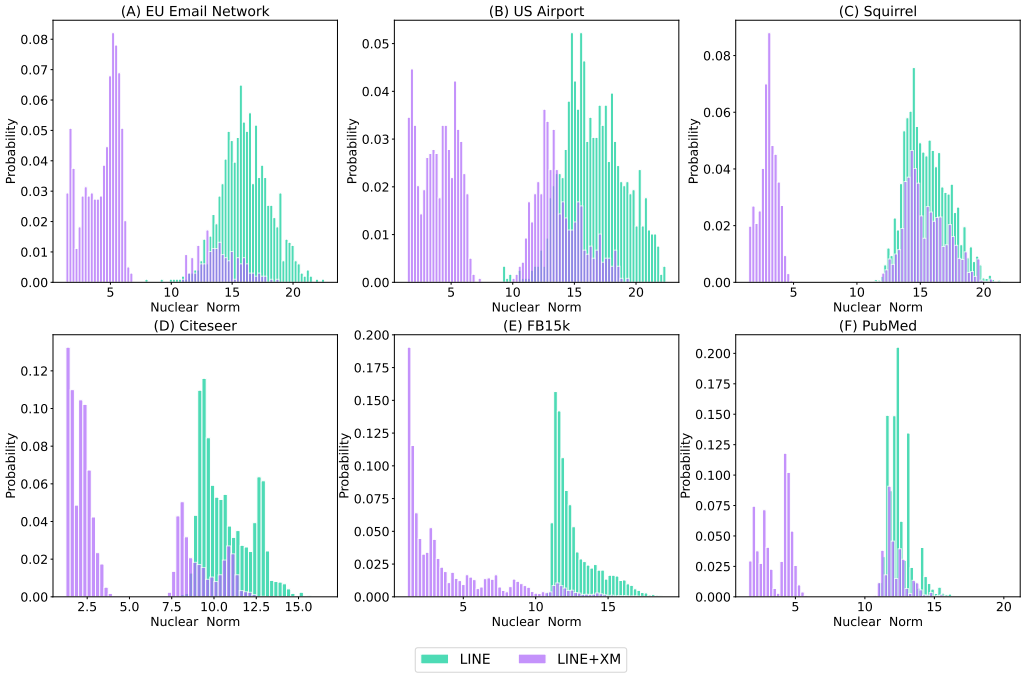


Figure 8: Nuclear norm distribution across the Explain matrices for LINE and LINE+XM. Observe how the distribution of LINE+XM is shifted to the left with a lower mean across each of the 6 datasets.

A.3 Runtime Analysis

We examine differences in run time per epoch (in seconds) between the original algorithms and the XM variants across different embedding dimensions. Figure 11 shows the run time comparison for the US Airport network, Figure 12 for the Squirrel network, Figure 13 for the Citeseer network, Figure 14 for the FB15k network, and Figure 15 for the PubMed network. We observe that the XM variants are comparable in runtime to their original versions.

A.4 Nuclear Norm Minimization

Figure 16a looks at all networks in Table 2 embedded using each of the 4 algorithms from Table 1. Results are averaged across 5 runs. We see that the nuclear norms of the Explain matrices are lower for the XM variants across each network. These differences are statistically significant with p-values less than 0.05. Table 16b lists the p-values.

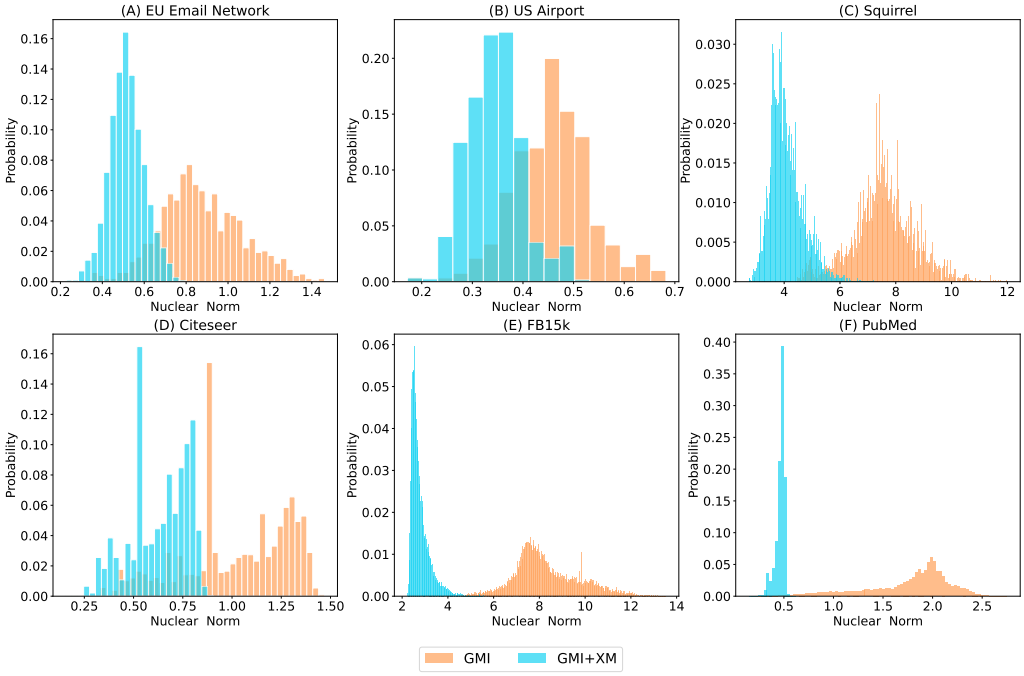


Figure 9: Nuclear norm distribution across the Explain matrices for DGI and DGI+XM. Observe how the distribution of DGI+XM is shifted to the left with a lower mean across each of the 6 datasets.

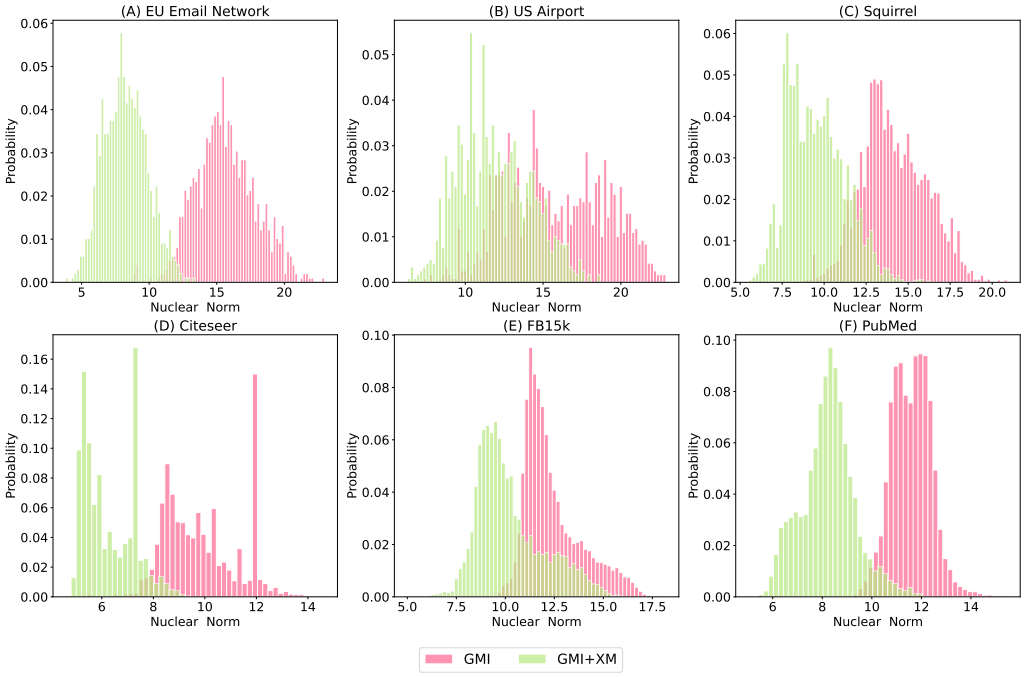


Figure 10: Nuclear norm distribution across the Explain matrices for GMI and GMI+XM. Observe how the distribution of GMI+XM is shifted to the left with a lower mean across each of the 6 datasets.

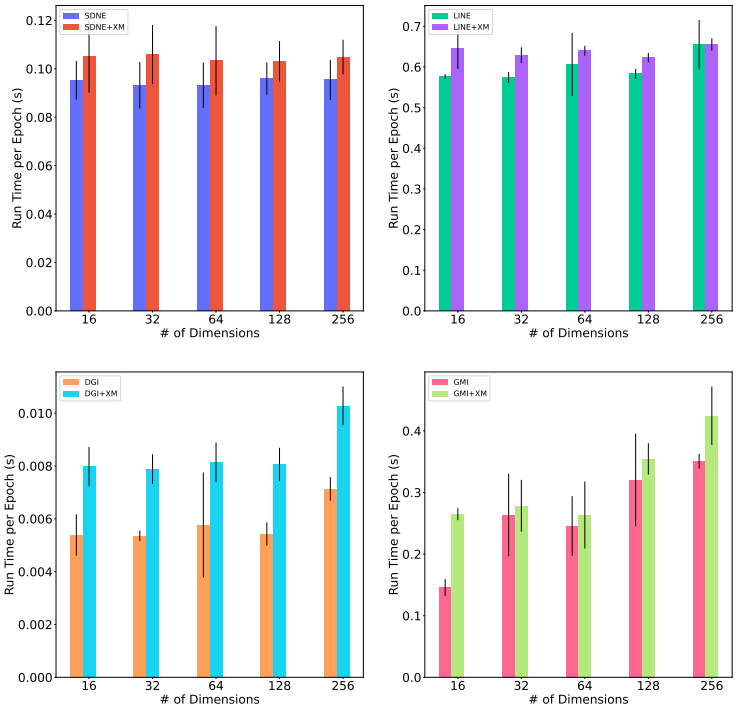


Figure 11: US Airport network: runtimes per epoch (in seconds) for SDNE, LINE, DGI, GMI, and their corresponding XM variants. Results are averaged across 5 runs. Observe that the runtimes per epoch are comparable to the original versions. Error bars show standard error.

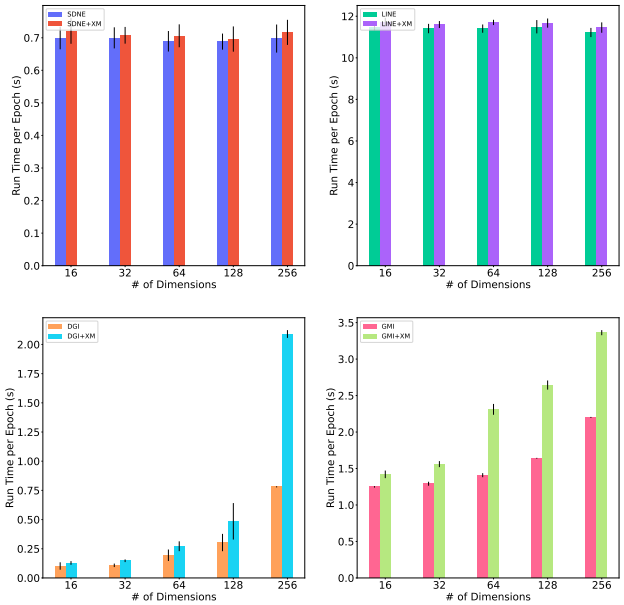


Figure 12: Squirrel network: runtimes per epoch (in seconds) for SDNE, LINE, DGI, GMI, and their corresponding XM variants. Results are averaged across 5 runs. Observe that the runtimes per epoch are comparable to the original versions. Error bars show standard error.

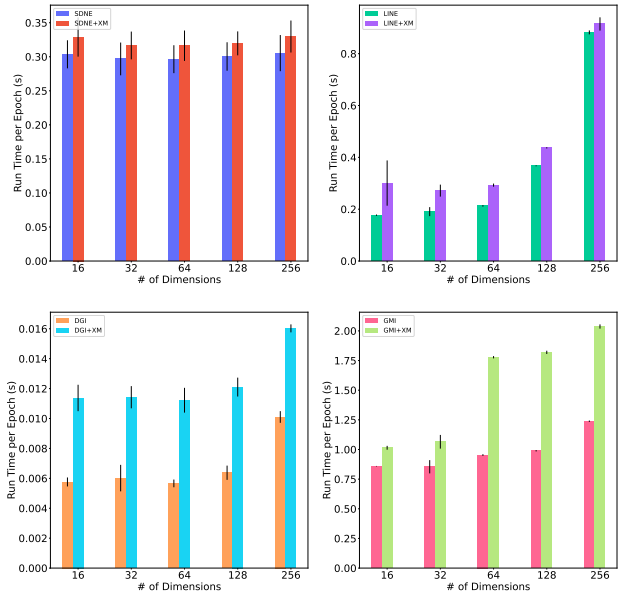


Figure 13: Citeseer network: runtimes per epoch (in seconds) for SDNE, LINE, DGI, GMI, and their corresponding XM variants. Results are averaged across 5 runs. Observe that the runtimes per epoch are comparable to the original versions. Error bars show standard error.

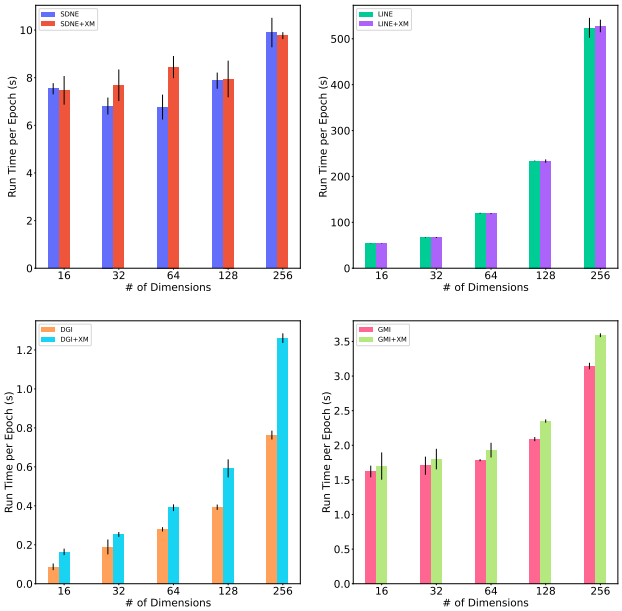


Figure 14: FB15k network: runtimes per epoch (in seconds) for SDNE, LINE, DGI, GMI, and their corresponding XM variants. Results are averaged across 5 runs. Observe that the runtimes per epoch are comparable to the original versions. Error bars show standard error.

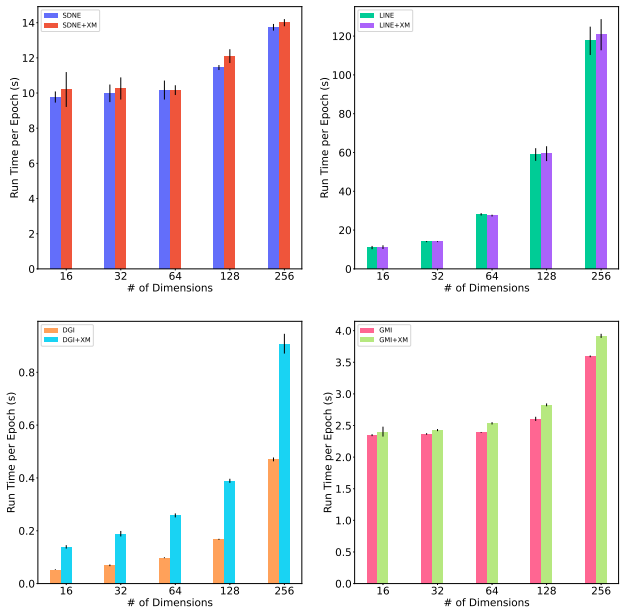
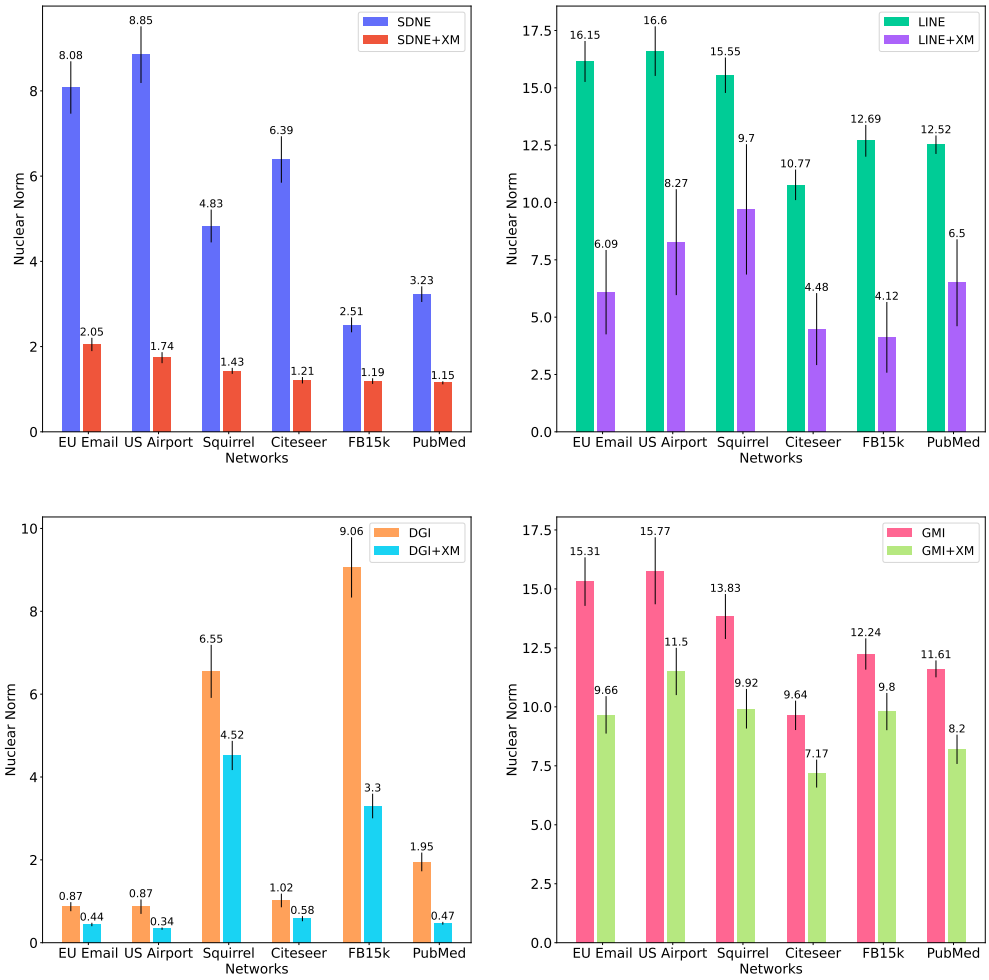


Figure 15: PubMed network: runtimes per epoch (in seconds) for SDNE, LINE, DGI, GMI, and their corresponding XM variants. Results are averaged across 5 runs. Observe that the runtimes per epoch are comparable to the original versions. Error bars show standard error.



(a) We average the nuclear norms of the Explain matrices of each node in the network. The experiment is repeated 5 times and average results are shown. Error bars depict standard error. Observe that the mean nuclear norms for the XM variants are lower than the original versions. The differences in means are statistically significant with p-values less than 0.05.

	EU Email	US Airport	Squirrel	Citeseer	FB15k	PubMed
SDNE vs. SDNE + XM	2.13e-10	9.78e-11	1.65e-10	7.76e-10	4.44e-11	3.96e-11
LINE vs. LINE + XM	1.96e-07	8.86e-06	6.72e-05	8.30e-06	1.38e-07	3.92e-06
DGI vs. DGI + XM	1.62e-03	1.47e-04	2.62e-04	4.88e-04	1.93e-07	5.03e-07
GMI vs. GMI + XM	1.80e-06	6.59e-06	3.81e-06	1.02e-07	2.88e-05	1.60e-07

(b) p-values associated with Figure 16a. The nuclear norms between the original versions and the XM variants are statistically significant.

Figure 16: Quantitative evaluation of the Explain matrices based on their nuclear norms across networks, original methods and their XM variants. The lower the norm, the better the Explain matrix.