DISENTANGLED AND SELF-EXPLAINABLE NODE REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Node representations, or embeddings, are low-dimensional vectors that capture node properties, typically learned through unsupervised structural similarity objectives or supervised tasks. While recent efforts have focused on explaining graph model decisions, the interpretability of *unsupervised* node embeddings remains underexplored. To bridge this gap, we introduce DISENE (**Di**sentangled and **Se**lf-Explainable **N**ode **E**mbedding), a framework that generates self-explainable embeddings in an unsupervised manner. Our method employs disentangled representation learning to produce dimension-wise interpretable embeddings, where each dimension is aligned with distinct topological structure of the graph. We formalize novel desiderata for disentangled and interpretable embeddings, which drive our new objective functions, optimizing simultaneously for both interpretability and disentanglement. Additionally, we propose several new metrics to evaluate representation quality and human interpretability. Extensive experiments across multiple benchmark datasets demonstrate the effectiveness of our approach.

1 INTRODUCTION

Self-supervised and unsupervised node representation learning (Hamilton, 2020) provide a powerful toolkit for extracting meaningful insights from complex networks, making them essential in modern AI and machine learning applications related to network analysis (Ding et al., 2024). These methods offer flexible and efficient ways to analyze high-dimensional networks by transforming them into low-dimensional vector spaces. This transformation enables dimensionality reduction, automatic feature extraction, and the use of standard machine learning algorithms for tasks such as node classification, clustering, and link prediction (Khosla et al., 2019). Furthermore, self-supervised node representations, or embeddings, enable visualization of complex networks and can be transferred across similar networks, enhancing understanding and predictive power in fields ranging from social networks to biological interactions.

Although widely adopted, unsupervised representation learning methods often face substantial challenges in terms of interpretability, necessitating complex and indirect approaches to understand what the learned embeddings actually represent (Piaggesi et al., 2024; Idahl et al., 2020; Gogoglou et al., 2019). This raises a critical question: *What information do these embeddings encode?*

While there has been a large body of work on explainable GNN models, limited attention has been given to embedding methods, which are the fundamental building blocks of graph based models. Existing efforts to explain embeddings are predominantly post-hoc (Piaggesi et al., 2024; Gogoglou et al., 2019; Khoshraftar et al., 2021; Dalmia et al., 2018) and heavily dependent on the initial embedding techniques used. Some approaches (Piaggesi et al., 2024) build upon existing literature for post-processing word embeddings (Subramanian et al., 2018; Chen & Zaki, 2017), focusing on minimizing reconstruction errors of node embeddings using over-complete auto-encoders to improve sparsity. Other works (Gogoglou et al., 2019; Khoshraftar et al., 2021; Dalmia et al., 2018) focus solely on extracting meaningful explanations without addressing the learning process itself.

We propose DISENE (**Di**sentangled and **Se**lf-Explainable **N**ode **E**mbedding), a framework that addresses these gaps by generating *self-explainable* unsupervised node embeddings. Our methodology is based on *disentangled* representation learning to encode node embeddings that are interpretable on a *dimension-wise* basis. Disentangled representation learning (Wang et al., 2022) aims to encode

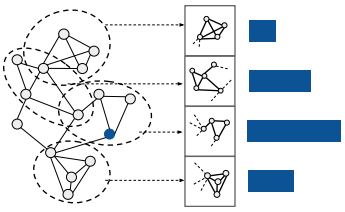

Figure 1: DISENE generates dimension-wise disentangled representations in which each embedding dimension is mapped to a mesoscale substructure in the input graph. The vector represents the embedding for the node marked in blue and the bars depict feature values.

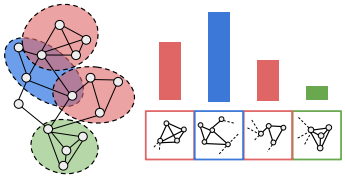

Figure 2: The overlap in dimension explanations aligns with the correlation between the node feature values for those dimensions. The dimension referenced by the blue subgraph shows a stronger correlation with the red dimensions and a lower correlation with the green dimension.

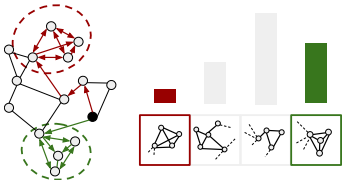

Figure 3: The node feature value indicates its proximity to the explanation substructure mapped to the corresponding dimension. The black node has a higher value for the dimension corresponding to the green subgraph (since it is 1 hop away) than for the dimension corresponding to the red subgraph (3 hops away).

latent variables that represent separate factors of variation in data, ensuring that each latent dimension corresponds to a distinct, independent aspect.

In graph data, node behaviour is strongly influenced by mesoscale structures like communities, which shape the network's organization and drive dynamics (Barrat et al., 2008). By leveraging disentangled representation learning, we capture these topological substructures more effectively, with each embedding dimension reflecting an independent graph unit (see Figure 1). We achieve this by introducing a new objective function to ensure *structural disentanglement*. Specifically, we optimize the embeddings so that each dimension is predictive of a unique substructure in the input graph. To avoid degenerate solutions, we incorporate an entropy-based regularizer which ensures that the mapped substructures are non-empty.

Our paradigm represents a shift in the language of explanations compared to the ones often considered when dealing with GNNs (Yuan et al., 2022). Explainability for GNNs often involves understanding which parts of the local computation graph (nodes, edges) and node attributes significantly influence the model's predictions (Funke et al., 2022; Ying et al., 2019; Schnake et al., 2021). On the other hand, the explanations that we aim to discover are inherently non-local, since they could involve mesoscale structures such as node clusters (Piaggesi et al., 2024), usually not included in the GNN computational graph.

To provide a comprehensive evaluation of the embeddings and uncover novel insights, we introduce several new metrics (for details refer to Section 3.3). For example, our *overlap consistency* measure (depicted in Figure 2) confirms that the physical overlap between the topological substructures identified as explanations aligns with the actual correlation of the corresponding embedding features or dimensions. This overlap offers insights into the interdependencies of node characteristics within the graph. Further, we attribute meaning to actual feature values by showing that the magnitude of node embedding entries corresponding to a dimension is correlated with the proximity (depicted in Figure 3) of the corresponding nodes to the "topological" subgraphs associated with that dimension. This measurement aids in understanding the closeness of nodes to different graph features, thereby enhancing spatial awareness within the graph structure.

To summarize we (i) formalize *new and essential criteria for achieving disentangled and explainable node representations*, offering a fresh perspective on explainability in unsupervised graph-based learning, (ii) introduce *novel evaluation metrics* to help quantifying the goodness of node representation learning in disentangled and explainable settings (iii) perform *extensive experimental analyses* to establish state-of-the-art results in self-explainable node feature learning. We release our code and data anonymously at this link.

2 PRELIMINARIES AND RELATED WORK

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, node embeddings are obtained through an encoding function $\mathbf{h} : \mathcal{V} \to \mathbb{R}^K$ that map each node to a points of a $K-$dimensional vector space $\mathbb{R}^K$, where typically

$D << |\mathcal{V}|$. We denote the K-dimensional embedding of a node $v \in \mathcal{V}$ as $\mathbf{h}(v) = [h_1(v), \ldots, h_K(v)]$, where $h_d(v)$ represents the value of the d-th feature of the embedding for node v. Alternatively, we can represent all node embeddings collectively as a matrix $\mathbf{H}(\mathcal{G}) \in \mathbb{R}^{V \times K}$, where each entry $\mathbf{H}_{vd} = h_d(v)$ corresponds to the d-th feature for node v. We can also refer to columns of such matrix, $\mathbf{H}_{:,d}$, as the dimensions of the embedding model space.

Node embeddings interpretability. Node embeddings are shallow encoding techniques, often based on matrix factorization or random walks Qiu et al. (2018). Since the latent dimensions in these models are not aligned with high-level semantics (Şenel et al., 2018; Prouteau et al., 2022), interpreting embeddings typically involves post-hoc explanations of their latent features (Gogoglou et al., 2019; Khoshraftar et al., 2021). Other works propose alternative methods to modify existing node embeddings, making them easier to explain with human-understandable graph features (Piaggesi et al., 2024; Shafi et al., 2024). From a different viewpoint, Shakespeare & Roth (2024) explore how understandable are the embedded distances between nodes. Similarly, Dalmia et al. (2018) investigate whether specific topological features are predictable, and then encoded, in node representations.

Graph neural networks interpretability. Graph Neural Networks (GNNs) (Wu et al., 2020) are deep models that operate via complex feature transformations and message passing. In recent years, GNNs have gained significant research attention, also in addressing the opaque decision-making process. Several approaches have been proposed to explain GNN decision process (Yuan et al., 2022), including perturbation approaches (Ying et al., 2019; Yuan et al., 2021; Funke et al., 2022), surrogate model-based methods (Vu & Thai, 2020; Huang et al., 2022), and gradients-based methods (Pope et al., 2019; Sanchez-Lengeling et al., 2020). In parallel, alternative research directions focused on concept-based explanations, i.e. high-level units of information that further facilitate human understandability (Magister et al., 2021; Xuanyuan et al., 2023; Magister et al., 2022).

Disentangled learning on graphs. Disentangled representation learning seeks to uncover and isolate the fundamental explanatory factors within data Wang et al. (2022). In recent years, these techniques have gained traction for graph-structured data (Liu et al., 2020; Li et al., 2021; Yang et al., 2020; Fan & Gao, 2024). For instance, FactorGCN (Yang et al., 2020) disentangles an input graph into multiple factorized graphs, resulting in distinct disentangled feature spaces that are aggregated afterwards. IPGDN (Liu et al., 2020) proposes a disentanglement using a neighborhood routing mechanism, enforcing independence between the latent representations as a regularization term for GNN outputs. Meanwhile, DGCL (Li et al., 2021) focuses on learning disentangled graph-level representations through self-supervision, ensuring that the factorized components capture expressive information from distinct latent factors independently.

3 Our Proposed Framework: DiSeNE

In this section, we begin by outlining the key desiderata for achieving disentangled and self-explainable node representations. Next, we design a novel framework that meets these objectives by ensuring that the learned node representations are both disentangled and interpretable. Finally, we introduce new evaluation metrics to effectively assess the quality of node representation learning in both disentangled and explainable settings.

3.1 Core Objectives and Desiderata

In the context of self-supervised graph representation learning, we argue that learning self-explainable node embeddings amounts to reconstructing the input graph in a human-interpretable fashion. Traditionally, dot-product models based on NMF (Yang & Leskovec, 2013) and LPCA (Chanpuriya et al., 2024) decompose the set of graph nodes into clusters, where each entry of the node embedding vector represents the strength of the participation of the node to a cluster. In this scenario, the dot-product of node embeddings becomes intuitively understandable, as it reflects the extent of shared community memberships between nodes–thereby providing a clear interpretation of edge likelihoods. This concept is also related to distance encoding methods (Li et al., 2020; Klemmer et al., 2023), where a node feature $h_d(u)$ is expressed as a function of the node's proximity $\zeta(u, \mathcal{S}_d) = \mathrm{AGG}(\{\zeta(u, v), v \in \mathcal{S}_d\})$ to the *anchor set* $\mathcal{S}_d \subset \mathcal{V}$, using an aggregation function AGG. Typically, distance encodings are

constructed by randomly sampling anchor sets (You et al., 2019), and used as augmented node features to enhance expressiveness and improve performance on downstream tasks.

Inspired by this idea, our goal is to optimize self-supervised node embeddings encoded by a GNN function $\mathbf{h} : \mathcal{V} \to \mathbb{R}^K$ trained on the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, such that node features resemble non-random, structurally meaningful anchor sets, thus improving human-interpretability. To achieve this, we propose three key desiderata for learning general-purpose node representations: *(i) structural faithfulness (ii)*, *dimensional interpretability*, and *(iii) structural disentanglement*. These desiderata serve as the foundational components of our approach, as detailed below.

Structural faithfulness. To implement the approach, we first train node embeddings in recovering the graph structure. We employ a random walk optimization framework based on the skip-gram model with negative sampling (Huang et al., 2021). The loss function for this framework is defined as:

$$\mathcal{L}_{\text{rw}} = -\sum_{(u,v)\sim P_{rw}} \log \sigma\big(\mathbf{h}(u)^\top \mathbf{h}(v)\big) - \sum_{(u',v)\sim P_n} \log \sigma\big(-\mathbf{h}(u')^\top \mathbf{h}(v)\big),$$

where $\sigma(\cdot)$ is the sigmoid function, P_{rw} is the distribution of node pairs co-occurring on random walks (positive samples), P_n is a distribution over randomly sampled node pairs (negative samples), and $\mathbf{h}(u)^\top \mathbf{h}(v)$ represents the dot product between the embeddings of nodes u and v. By optimizing this loss function, we encourage nodes that co-occur in random walks to have similar embeddings, effectively preserving the graph's structural information in the embedding space.

Dimensional interpretability. Given that our embeddings are structurally faithful–meaning they effectively encode the input graph's structure–we should be able to interpret each embedding dimension in terms of the graph's structural properties. We achieve this by attributing local subgraphs to different latent dimensions. Consider the likelihood of an edge (u, v), defined as $\hat{y}(u, v; \mathbf{h}) = \sigma\Big(\sum_{d=1}^{K} h_d(u) h_d(v)\Big)$. To understand how each dimension d contributes to this likelihood, we compute the edge-wise dimension importance $\phi_d(u, v; \mathbf{h})$ as the deviation of the dimension-specific contribution from its average over all edges:

$$\phi_d(u, v; \mathbf{h}) = h_d(u) h_d(v) - \frac{1}{|\mathcal{E}|} \sum_{(u',v')\in\mathcal{E}} h_d(u') h_d(v'). \tag{1}$$

Since the dot-product is a linear function $\sum_{d=1}^{K} \alpha_d h_d(u) h_d(v) + \beta$ with unitary coefficients $\alpha_d \equiv 1$ and zero intercept $\beta \equiv 0$, Eq. (1) corresponds to the formulation of LinearSHAP attribution scores (Lundberg & Lee, 2017), using the set of training edges as the background dataset. Essentially, the attribution function $\phi_d(u, v; \mathbf{h})$ indicates whether a specific dimension d contributes positively to an edge's likelihood. A positive attribution score means that the dimension increases the likelihood of predicting the edge. Leveraging this property, we generate dimension-wise explanations for the latent embedding model by collecting edge subsets with positive contributions:

$$\mathcal{E}_d = \{(u, v) \in \mathcal{E} : \phi_d(u, v; \mathbf{h}) > 0\}. \tag{2}$$

These self-explanations take the form of global edge masks $\mathbf{M}^{(d)} \in \mathbb{R}_{\geq 0}^{|\mathcal{V}|\times|\mathcal{V}|}$, where each entry is defined as $M_{uv}^{(d)}(\phi_d; \mathbf{h}) = \max\{0, \phi_d(u, v; \mathbf{h})\}$. By applying these masks to the adjacency matrix $\mathbf{A}$ through Hadamard product ($\odot$), we obtain $\mathbf{A}^{(d)} = \mathbf{A} \odot \mathbf{M}^{(d)}$. Each masked adjacency matrix $\mathbf{A}^{(d)}$ highlights the subgraph associated with dimension d. From these masked adjacency matrices, we construct edge-induced subgraphs $\mathcal{G}_d = (\mathcal{V}_d, \mathcal{E}_d)$, where $\mathcal{V}_d$ is the set of nodes involved in edges $\mathcal{E}_d$. These subgraphs act as anchor sets for the model, providing interpretable representations of how each embedding dimension relates to specific structural patterns within the graph. We will refer to edge-induced subgraphs computed as the aforementioned procedure (pseudo-code in Appendix A.5) as *explanation subgraphs/substructures* or *topological components* of the embedding model.

Structural disentanglement. To enhance the effectiveness of structurally faithful encodings, each dimension of the latent space should encode an *independent* structure of the input graph, effectively acting as an anchor subgraph. Inspired by community-affiliation models (Yang & Leskovec, 2013; 2012), we introduce a node affiliation matrix $\mathbf{F} \in \mathbb{R}^{|\mathcal{V}|\times K}$ that captures the association between each node $u \in \mathcal{V}$ and anchor subgraph $\mathcal{G}_d = (\mathcal{V}_d, \mathcal{E}_d)$. Specifically, each entry $\mathbf{F}_{ud}$ is proportional to the

magnitude of predicted meaningful connections between node u and other nodes in $\mathcal{G}_d$, expressed using the per-dimension attribution scores from Eq. (1):

$$\mathbf{F}_{ud}(\{\mathbf{h}\}) = \sum_{v \in \mathcal{V}_d} \phi_d(u, v; \mathbf{h}) \tag{3}$$

This aggregates the contributions of dimension d to the likelihood of edges involving node u. To achieve structure-aware disentanglement, we enforce soft orthogonality among the columns of the affiliation matrix[1]. This ensures that different embedding dimensions capture independent structures, leading to nearly non-overlapping sets of predicted links for each dimension. We express the columns of the affiliation matrix as $\mathbf{F}_{:,d}$ and obtain the disentanglement loss function as:

$$\mathcal{L}_{\text{dis}} = \sum_{d=1}^{K} \sum_{l=1}^{K} \left[\cos\left(\mathbf{F}_{:,d}, \mathbf{F}_{:,l}\right) - \delta_{d,l}\right] \tag{4}$$

where $\cos\left(\mathbf{F}_{:,d}, \mathbf{F}_{:,l}\right)$ denotes the cosine similarity between the d-th and the l-th columns of $\mathbf{F}$, and and $\delta_{d,l}$ is the Kronecker delta function (1 if $d = l$, 0 otherwise). This approach enables us to obtain disentangled representations (Wang et al., 2022), where each dimension corresponds to a single underlying structural factor in the graph. Although it is possible to obtain disentanglement at more coarse-grained levels (e.g., with groups of dimensions), we focus on single-feature disentanglement because it inherently leads to dimension-wise interpretability.

3.2 Our approach: DiSeNE

Building upon the above components, introduced to satisfy our desiderata for interpretable node embeddings, we present our approach, DiSeNE. Specifically, DiSeNE takes as input the raw node attributes $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times F}$ and, depending on the encoder architecture, also the adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$. The input is encoded into an intermediate embedding layer $\mathbf{Z} \in \mathbb{R}^{|\mathcal{V}| \times D}$. Next, DiSeNE processes the embedding matrix $\mathbf{Z}$ to compute the likelihood of link formation between node pairs, given by $\hat{y}(u, v; \mathbf{h}) = \sigma(\mathbf{h}(u)^\top \mathbf{h}(v))$ where $\mathbf{h}(v) = \rho(\mathbf{W}^\top \mathbf{z}(v))$ are the final node representations in $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times K}$, obtained by applying a linear transformation $\mathbf{W} \in \mathbb{R}^{D \times K}$ followed by a non-linear activation function ρ. To encode $\mathbf{z}$, we employ architectures incorporating fully connected layers and graph convolutional layers (Wu et al., 2019). This process can be further enhanced by integrating more complex message-passing mechanisms or MLP operations. For example, the message-passing could initiate from an MLP-transformed node attribute matrix, MLP($\mathbf{X}$), or incorporate more sophisticated architectures beyond simple graph convolutions for increased expressiveness (Xu et al., 2018; Veličković et al., 2017).

The embeddings are optimized by combining the previously described objective functions for preserving structural faithfulness and achieving structural disentanglement, thereby improving dimensional interpretability. To avoid degenerate disentanglement solutions–specifically, the emergence of "empty" clusters characterized by near-zero columns in $\mathbf{F}$ that, while orthogonal to others, fail to convey meaningful information–we introduce a regularization strategy. This regularization ensures a minimal but significant level of connectivity within each topological substructure. Specifically, we enforce that the total amount of predicted edges in each anchor subgraph $\mathcal{G}_k$ - $\sum_{u,v \in \mathcal{V}} \phi_k(u, v; \mathbf{h})$ - to be non-zero. We found a more stable and precise approach by enforcing that the aggregated node features of each embedding dimension are non-zero, achieved by maximizing the entropy:

$$\mathcal{H} = -\sum_{d=1}^{K} \left(\frac{\sum_u h_d(u)}{||\sum_u \mathbf{h}(u)||_1} \right) \log \left(\frac{\sum_u h_d(u)}{||\sum_u \mathbf{h}(u)||_1} \right).$$

Thus, the model is optimized by minimizing the following comprehensive loss function:

$$\mathcal{L} = \mathcal{L}_{\text{rw}} + \mathcal{L}_{\text{dis}} + \lambda_{\text{ent}} \left(1 - \frac{\mathcal{H}}{\log K}\right)$$

The hyperparameter λ_{ent} determines the strength of the regularization, controlling the stability for explanation subgraph sizes across the various latent dimensions. We report the pseudo-code of DiSeNE in Appendix A.3, along with a space-time complexity analysis.

[1]Note that $\mathbf{F}_{ud}(\{\mathbf{h}\}) \equiv \sum_{v \in \mathcal{V}} h_d(u) h_d(v) - \frac{|\mathcal{V}|}{|\mathcal{E}|} \sum_{(u',v') \in \mathcal{E}} h_d(u') h_d(v')$ and assuming $|\mathcal{V}| << |\mathcal{E}|$, the second term becomes negligible, allowing us to approximate $\mathbf{F}_{ud}(\{\mathbf{h}\})$ and reduce computational costs.

3.3 PROPOSED EVALUATION METRICS

In the following, we introduce novel metrics to quantify multiple aspects related to interpretability and disentanglement in node embeddings, which we use to compare models in our experiments.

Comprehensibility. Comprehensibility measures how closely the identified topological explanations align with ground-truth clusters, which are crucial in the evolution of numerous complex real-world systems, i.e. community modules (Girvan & Newman, 2002; Hric et al., 2014). We evaluate comprehensibility by treating edges in explanation masks $\{\mathbf{M}^{(d)}\}_{d=1,\dots,K}$ as retrieved items from a query, and measuring their overlap with the edges in the ground-truth communities using precision, recall, and F_1-score. Let $\mathcal{C}(\mathcal{E}) = \{\mathcal{C}^{(1)}, \dots, \mathcal{C}^{(m)}\}$ denotes the set of truthful link communities of the input graph[2]. Associated to partition $\mathcal{C}^{(i)}$, we define ground-truth edge masks $\mathbf{C}^{(i)} \in \{0,1\}^{V\times V}$ with binary entries $C^{(i)}_{uv} = [(u,v) \in \mathcal{C}_i]$. *Comprehensibility* score is given by the maximum F_1-score across ground-truth index:

$$\mathrm{Comp}(\mathbf{M}^{(d)}) = \max_i \left\{ F_1(\mathbf{M}^{(d)}, \mathbf{C}^{(i)}) \right\} = \max_i \left\{ \frac{2}{\mathrm{prec}(\mathbf{M}^{(d)}, \mathbf{C}^{(i)})^{-1} + \mathrm{rec}(\mathbf{M}^{(d)}, \mathbf{C}^{(i)})^{-1}} \right\} \tag{5}$$

For precision, we weigh relevant item scores with normalized embedding masks values: $\mathrm{prec}(\mathbf{M}^{(d)}, \mathbf{C}^{(i)}) = \frac{\sum_{u,v} M^{(d)}_{uv} C^{(i)}_{uv}}{\sum_{u,v} M^{(d)}_{uv}}$. For recall, we weigh binarized embedding masks values with normalized ground-truth scores[3]: $\mathrm{rec}(\mathbf{M}^{(d)}, \mathbf{C}^{(i)}) = \frac{\sum_{u,v} [M^{(d)}_{uv} > 0] C^{(i)}_{uv}}{|\mathcal{C}^{(i)}|}$.

Sparsity. We refer to sparsity as the amount of disorder in the dimension's explanations, it is generally defined as the ratio of the number of bits needed to encode an explanation compared to those required to encode the input (Funke et al., 2022). Given that concise explanations are more effective in delivering clear insights, enhancing human understanding, we evaluate *sparsity* by measuring the normalized Shannon entropy over the mask distribution:

$$\mathrm{Sp}(\mathbf{M}_d) = -\frac{1}{\log|\mathcal{E}|} \sum_{(u,v)\in\mathcal{E}} \left(\frac{M^{(d)}_{uv}}{\sum_{u',v'} M^{(d)}_{u'v'}} \right) \log \left(\frac{M^{(d)}_{uv}}{\sum_{u',v'} M^{(d)}_{u'v'}} \right). \tag{6}$$

A lower entropy in the mask distribution indicates higher sparsity.

Overlap Consistency. In explaining latent space representations, it is essential to comprehend how input space factors influence specific latent features. A well-structured, disentangled latent space should correspond to distinct, uncorrelated topological structures. We aim to quantify how different topological components affect pairwise feature correlations in the latent space. To achieve this, we propose a metric that measures the strength of association between the physical overlap of the explanation substructures $\{\mathcal{G}_d\}$ and the correlation among corresponding latent dimensions $\{\mathbf{H}_{:,d}\}$. We compute the overlap between two subgraph components using the Jaccard Similarity Index (JSI) of their edge sets from Eq. (2): $\mathrm{JSI}(d,l;\mathbf{h}) = \frac{|\mathcal{E}_d \cap \mathcal{E}_l|}{|\mathcal{E}d \cup \mathcal{E}_l|}$. The *overlap consistency* (OvC) metric measures the linear correlation between the pairwise JSI values and the squared Pearson correlation coefficients (ρ^2) of the embedding features:

$$\mathrm{OvC}(\mathbf{h}) = \rho\Big([\mathrm{JSI}(d,l;\mathbf{h})]_{d<l}, [\rho^2(\mathbf{H}_{:,d}, \mathbf{H}_{:,l})]_{d<l} \Big) \tag{7}$$

where $[*]_{d<l}$ denotes the condensed list of pair-wise similarities. By using ρ^2 we remain agnostic about the sign of the correlation among latent features, since high overlaps could originate from both cases. This metric provides a systematic way to assess the extent to which topological partitions align with the distribution of features in the embedding space, thus offering deeper insights into the interpretability and disentanglement of the learned representations.

[2]Synthetic graphs can be constructed with ground-truth relevant sub-structures (like BA-SHAPES (Ying et al., 2019) or SBM graphs). In real-world graphs, it is usually reasonable to assume that the community structure (Fortunato, 2010) can serve as ground-truth.

[3]For the precision, we normalize with the sum of scores because they are continuous. For recall, we use the cardinality in place of the sum because the ground-truth has binary scores.

Positional Coherence. In unsupervised node embeddings, it is crucial to assess how well the latent space captures the graph's structure by encoding the positional relationships of nodes. An effective representation should preserve meaningful spatial properties that reflect node proximity and connectivity patterns. To achieve this, we propose to measure the extent to which node entries reflect their positions within the graph. Typically, positional encoding (Lu et al., 2021; Li et al., 2020; You et al., 2019) involves the use of several sets of node *anchors* $\mathcal{S}_d \subset \mathcal{V}$ that establish an intrinsic coordinate system. This system influences the node u's features based on the node's proximity $\zeta(u, \mathcal{S}_d) = \mathrm{AGG}(\{\zeta(u, v), v \in \mathcal{S}_d\})$, where AGG denotes a specific pooling operation. As node proximity, we used the inverse of the shortest path distance $\zeta_{spd}(u, v) \equiv (1 + d_{spd}(u, v))^{-1}$. As the anchor sets, we chose the embedding substructures used for explanations, $\mathcal{S}_d \equiv \mathcal{V}_d$.

For a specified pair of dimensions (d, l), we assess the correlation between node features along dimension d and the corresponding distances to the topological component indexed by l via Feature-Proximity Correlation: $\mathrm{FPC}(d, l; \mathbf{h}) = \rho\Big(\big[\zeta_{spd}(u, \mathcal{V}_d)\big]_{u\in\mathcal{V}}, \mathbf{H}_{:,l}\Big)$. The *positional coherence* metric (PoC) is defined to specifically evaluate the degree to which each feature d is uniquely correlated with its corresponding topological component $\mathcal{V}_d$, without being significantly influenced by correlations with other substructures. This metric is calculated as the ratio of the average FPC for the given dimensions to the average FPC computed with pairs of permuted dimensions:

$$\mathrm{PoC}(\mathbf{h}) = \frac{\sum_d \mathrm{FPC}(d, d; \mathbf{h})}{\big\langle \sum_d \mathrm{FPC}(d, \pi(d); \mathbf{h}) \big\rangle_\pi} \tag{8}$$

where $\langle . \rangle_\pi$ denotes an empirical average over multiple permutations. By comparing with random feature-subgraph pairs, the metric avoid promoting models with redundancies in the latent features, where high correlations with other topological components are possible.

Plausibility. Given the importance of node representations in downstream tasks, it is crucial to assess whether the features influencing predictions align with human expectations using synthetic benchmarks. To this end, we construct instance-wise explanations to determine if key features correspond to the topological structures behind the ground-truth labels. Typical post-hoc explainers (Bodria et al., 2023) that produce feature importance scores often fail in this context because node embeddings are inherently uninterpretable, leading to useless explanations. Our approach overcomes this limitation by mapping feature importance back to the graph's structural components, enabling a more meaningful evaluation of how well the embeddings capture the underlying factors driving node behavior.

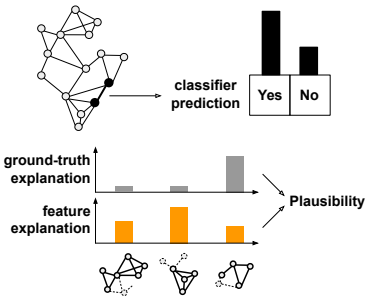

Figure 4: Schematic view of Plausibility metric computation. A high plausibility score indicates that the dimensions deemed more comprehensible also received higher importance scores from the post-hoc feature attribution technique.

Consider training a downstream binary classifier $b : \mathbb{R}^K \to [0, 1]$, such as for node classification or link prediction. We detail the procedure for link prediction here (see also Figure 4), but we will also report the methodology for node classification in Appendix A.6. For an edge instance $\mathbf{h}(u, v)$ (which could be derived from node-pair operations such as $\mathbf{h}(u) \odot \mathbf{h}(v)$), we employ post-hoc methods to determine the feature relevance for the classifier prediction on the node pair instance (u, v), $\{\Psi_j(u, v; b)\}_{j=1,\dots,K}$ that outputs important score for each of the embedding dimensions. Similarly to scores in Eq. (1), we define task-based masks $\mathbf{B}^{(j)} \in \mathbb{R}^{V\times V}_{\geq 0}$ which aggregate the individual logics of the classifier predictions: $B^{(j)}_{uv}(\Psi_j; b) = \max\{0, \Psi_j(u, v; b)\}$. To evaluate the consistency of a prediction, we consider the F_1-score in Eq. (5) related to the ground-truth structure of the edge under study, indexed by $g(u, v)$. Specifically, we define *plausibility* for an individual prediction as the average comprehensibility relative to the instance ground-truth, weighted by the computed feature importance:

$$P\ell(u, v; b) = \frac{\sum_{j=1}^K f(\Psi_j(u, v; b)) F_1(\mathbf{B}^{(j)}, \mathbf{C}^{g(u,v)})}{\sum_{j=1}^K f(\Psi_j(u, v; b))} \tag{9}$$

where f is a function guaranteeing the non-negativity of relevance weights. This ensures that only the features that are both interpretable and significant to the local prediction contribute substantially to the score, penalizing instead those important features that are not human-comprehensible.

4 EXPERIMENTS

We conduct large-scale experiments to answer the following research questions.

- **Human understandability:** How comprehensible and sparse are the explanation substructures generated by DISENE?
- **Structural disentanglement:** Do the disentangled subgraphs reveal intrinsic properties of node embeddings, like feature correlations and latent positions?
- **Utility for downstream tasks:** Are the identified substructures plausible and coherent enough to serve as explanations in downstream tasks?

To address our research questions, we extract topological components of multiple embedding methods, trained on different graph-structured data, by computing edge subsets defined in Eq (2), and analyzing embedding metrics defined in Section 3.3. In the following sections, we describe the data, models, the experimental setup and results from the comparison.

4.1 DATASETS AND COMPETITORS

Datasets. We ran experiments on four real-world datasets (CORA, WIKI, FACEBOOK, PPI), and six synthetic datasets (RING-OF-CLIQUES, SBM, BA-CLIQUES, ER-CLIQUES, TREE-CLIQUES and TREE-GRIDS) containing 32 cliques with 10 nodes as ground-truth motifs. Statistics for these datasets are provided in Table A1 in the Appendix. Additionally we employ several biological datasets (see Appendix A.4) for the evaluation on multi-label node classification). BA-CLIQUES and ER-CLIQUES are variations of the BA-SHAPES (Ying et al., 2019) where we randomly attach cliques –instead of house motifs– to Barabási-Albert and Erdős-Rényi random graphs. TREE-CLIQUES and TREE-GRIDS (Ying et al., 2019) are composed of a base balanced tree, with cliques and 3x3 grid motifs respectively. RING-CLIQUES and SBM are implemented in NetworkX[4]. For synthetic data, we present only results for plausibility metrics, leaving the other findings in the Appendix A.5.

Methods. We compare different node embedding methods. Competitors include DEEPWALK (Perozzi et al., 2014), Graph Autoencoder (GRAPHAE) (Salha et al., 2021), INFWALK (Chanpuriya & Musco, 2020), and GRAPHSAGE (Hamilton et al., 2017). Moreover, we apply the DINE retrofitting approach (Piaggesi et al., 2024) to post-process embeddings from DEEPWALK and GRAPHAE. We evaluate our method DISENE in two variants: a 1-layer fully-connected encoder (DISE-FCAE) and a 1-layer convolutional encoder (DISE-GAE). GNN-based methods are trained using the identity matrix as node features. Details on hyperparameters and training settings are provided in the Appendix A.2.

Setup. In experiments on real-world graphs, we investigate latent space interpretability and disentanglement metrics by keeping the output embedding dimension fixed at 128. This dimensionality was chosen to ensure that all methods achieve optimal performance in terms of test accuracy, specifically for link prediction (see the Appendix A.4 for extensive downstream task results). For synthetic data, since we investigated plausibility metric referred to a downstream classifier, thus we did not focus on a specific dimension but we selected the best score metric varying the output dimensions in the list $[2, 4, 8, 16, 32, 64, 128]$. Each reported result is an average over 5 runs. For link prediction, we use a 90%/10% train/test split, and for node classification, we use an 80%/20% split. All results refer to the training set, except for downstream task experiments, where we present results for the test set.

4.2 RESULTS AND DISCUSSION

Are the topological substructures both comprehensible and sparse to support human understandability? Here we explore how well the represented topological structures can serve as global explanations for the node embeddings, quantifying the Comprehensibility in the terms of associations between model parameters and human-understandable units of the input graph, as well as the Sparsity of these associations. In Table 1 we show compact scores as the average values $\frac{1}{K}\sum_{d=1}^{K}\mathrm{Comp}(\mathbf{M}_d)$ and $1-\frac{1}{K}\sum_{d=1}^{K}\mathrm{Sp}(\mathbf{M}_d)$ over all the embedding features. For sparsity we report the value subtracted from 1 to have all the scores better with higher values.

[4] https://networkx.org/documentation/stable/reference/generators.html

Table 1: Comprehensibility and sparsity results for real-world datasets. Best scores are in bold, second best results are underlined.

Method	Comprehensibility				Sparsity			
	CORA	WIKI	FB	PPI	CORA	WIKI	FB	PPI
DEEPWALK	.363±.003	.356±.002	.602±.004	.281±.002	.183±.001	.165±.002	.130±.004	.136±.003
GRAPHAE	.299±.001	.248±.002	.481±.014	.263±.003	.182±.002	.164±.004	.154±.003	.135±.003
INFWALK	.281±.002	.288±.001	.658±.006	.312±.002	.211±.003	.185±.002	**.318**±.010	.177±.002
GRAPHSAGE	.358±.004	.307±.007	.583±.003	.306±.005	.189±.001	.189±.003	.145±.002	.172±.001
DW+DINE	.511±.051	.496±.014	.813±.025	**.569**±.022	.317±.036	.266±.007	.226±.009	.188±.002
GAE+DINE	.569±.004	.591±.004	.843±.005	.484±.007	.290±.001	.252±.001	.195±.002	.198±.002
DISE-FCAE	.822±.001	.755±.003	**.971**±.001	.484±.001	**.504**±.001	**.419**±.001	.297±.002	**.282**±.001
DISE-GAE	**.834**±.003	**.762**±.004	.967±.001	.515±.001	.496±.001	.418±.001	.304±.003	.254±.002

DEEPWALK and INFWALK show moderate performance in **Comprehensibility**, excelling slightly on FB but underperforming on PPI, while GRAPHAE consistently lags behind, particularly on WIKI and PPI. GRAPHSAGE shows good comprehensibility across CORA and FB. Incorporating DINE improves results, especially for GAE+DINE, which achieves improved scores on all datasets. The proposed models, DISE-FCAE and DISE-GAE, deliver the highest overall performance. DISE-FCAE performs well on FB, while DISE-GAE excels across CORA and WIKI. However, both models show sub-optimal results on PPI, suggesting potential for further improvement on this dataset.

DEEPWALK and GRAPHAE offer moderate **Sparsity**, peaking on CORA, but underperform on other datasets. INFWALK excels on FB but shows moderate results elsewhere, while GRAPHSAGE performs poorly in terms of sparsity across all datasets. DEEPWALK and GAE significantly improve their sparsity with DINE, particularly on CORA. For the proposed models, DISE-FCAE performs best across datasets CORA, WIKI and PPI. Meanwhile, DISE-GAE obtains the highest value on FB.

Can the identified subgraphs explain the intrinsic characteristics of the node embeddings? Here we explore how well the defined topological units represent information in the node embedding space, providing insights into how the relative and absolute positioning of topological structures influences the feature encoding within a graph. By quantifying these relationships, we can better understand the underlying patterns and structural information in the graph, potentially leading to more robust and interpretable models. In Table 2 we report Positional Coherence and Overlap Consistency for the examined embedding methods. For the second metric, as node proximity we used the inverse of the shortest path distance with `sum` as pooling.

Table 2: Overlap consistency and positional coherence results for real-world datasets. Best scores are in bold, second best result are underlined.

Method	Overlap Consistency				Positional Coherence			
	CORA	WIKI	FB	PPI	CORA	WIKI	FB	PPI
DEEPWALK	.137±.009	.143±.006	.115±.007	.015±.003	1.078±0.025	0.835±0.025	1.119±0.015	1.009±0.015
GRAPHAE	.269±.002	.295±.004	.273±.017	.452±.008	1.023±0.006	1.040±0.002	1.001±0.013	1.016±0.001
INFWALK	.008±.003	.023±.002	.021±.002	.134±.002	1.004±0.011	0.998±0.004	0.938±0.053	0.999±0.002
GRAPHSAGE	.211±.003	.136±.017	.230±.007	.097±.040	1.099±0.012	1.103±0.010	1.005±0.007	1.018±0.002
DW+DINE	**.900**±.012	.804±.032	.851±.017	.855±.016	1.790±0.076	2.126±0.065	1.792±0.058	1.247±0.043
GAE+DINE	.560±.010	.610±.006	.801±.016	.646±.003	2.317±0.028	2.551±0.048	1.783±0.037	1.098±0.004
DISE-FCAE	.885±.008	**.863**±.006	**.939**±.030	.548±.001	5.210±0.080	3.540±0.082	3.348±0.085	**1.283**±0.004
DISE-GAE	.853±.007	.811±.008	.889±.007	**.887**±.004	**5.300**±0.193	**4.343**±0.144	**3.388**±0.054	1.261±0.005

DEEPWALK and INFWALK perform poorly for **Overlap Consistency**, while GRAPHAE shows moderate scores, particularly on PPI. GRAPHSAGE performs slightly worse, with the best overlap consistency on FB and CORA. DW+DINE achieves strong scores across all datasets, while GAE+DINE performs solidly but slightly lower, with its best result on FB. The proposed models, DISE-FCAE and DISE-GAE, outperform all others, achieving the highest consistency across all datasets except on CORA. DISE-FCAE excels on FB and WIKI, while DISE-GAE achieves the best overall score on PPI.

Table 3: Plausibility results for synthetic datasets with best scores in bold, second best underlined.

Method	Link Prediction				Node Classification			
	RING-CL	SBM	BA-CL	ER-CL	BA-CL	ER-CL	TR-CL	TR-GR
DEEPWALK	.234±.003	.205±.008	.173±.002	.160±.006	.146±.002	.141±.003	.103±.007	.091±.002
GRAPHAE	.183±.003	.160±.002	.145±.004	.145±.005	.130±.002	.135±.006	.083±.001	.072±.001
INFWALK	.224±.005	.181±.005	.218±.007	.212±.008	.129±.002	.141±.004	.097±.002	.093±.004
GRAPHSAGE	.252±.005	.217±.003	.186±.006	.178±.005	.160±.004	.154±.002	.093±.002	.084±.003
DW+DINE	.943±.012	.904±.002	.744±.008	.724±.040	.320±.031	.327±.008	.549±.015	.627±.004
GAE+DINE	.549±.005	.547±.014	.418±.011	.387±.002	.351±.011	.397±.003	.366±.013	.254±.005
DISE-FCAE	**.978**±.001	**.924**±.006	**.950**±.006	.938±.014	**.820**±.011	.791±.012	**.860**±.004	**.810**±.008
DISE-GAE	.969±.002	.910±.006	.936±.003	**.941**±.005	.813±.003	**.797**±.009	.791±.005	.800±.004

DEEPWALK, GRAPHAE, and GRAPHSAGE demonstrate moderate **Positional Coherence**. INFWALK consistently scores around 1.0 on all datasets, indicating stable but unremarkable coherence. Incorporating DINE leads to substantial improvements for both DEEPWALK and GAE, achieving notable gains on CORA, WIKI and FB. The proposed models, DISE-FCAE and DISE-GAE, far outperform other methods, with DISE-FCAE achieving top scores on PPI, while DISE-GAE dominates on CORA, WIKI and FB (though with higher variance): both models show consistent superiority.

Are the identified latent structures sufficiently meaningful to serve as explanations for downstream tasks? Node embeddings serve as versatile feature representations suitable for downstream tasks, though they typically function as "tabular-like" feature vectors without semantic labels for each feature. This limitation restricts the use of established post-hoc analysis methods (Bodria et al., 2023) like LIME, SHAP, etc. Our method allows us to link topological substructures with embedding features, thereby assigning semantic labels to node vectors. Consequently, we are able to explain a downstream classifier trained with unsupervised embeddings using feature attribution. Our goal is to assess whether the task-important features align with human understanding by measuring the Plausibility.

In these experiments we consider node classification and link prediction as binary downstream tasks, training a logistic regression classifier $b(x;\boldsymbol{\beta}) = \sigma(\sum_{j=1}^{K} \beta_j h_j(x) + \beta_0)$, where x is a node/link instance. We use SHAP (Lundberg & Lee, 2017) to compute the instance-wise feature attribution values $\{\Psi_j(x;b)\}_{j=1\ldots K}$. For node classification, we consider positive instances as the nodes inside a clique in the synthetic graph. Accordingly, the ground-truth explanation for a node is the clique it belongs to. For link prediction, we focus on test edges that were inside a clique before removal, where the ground-truth explanation is again the clique itself. We used $f(*) = \max(0, *)$ as non-negative weighting function for computing plausibility scores.

Table 3 shows average **Plausibility** scores for downstream tasks over test instances with correct predicted label. DEEPWALK, GRAPHAE, and INFWALK perform modestly, with DEEPWALK scoring the highest among these on RING-CL and INFWALK showing relative strength on BA-CL. GRAPHSAGE significantly underperforms across all tasks, especially in node classification. The addition of DINE improves both DEEPWALK and GAE. DW+DINE excels with strong performance on RING-CL, SBM, and TREE datasets, while GAE+DINE achieves slightly worst results, particularly on node classification tasks, such as in TR-GR. Within the proposed models, DISE-FCAE and DISE-GAE consistently achieve the highest scores ranking as the best two methods overall.

5 CONCLUSIONS

We present DISENE, a novel framework for generating self-explainable unsupervised node embeddings. To build our framework, we design new objective functions that ensure structural faithfulness, dimensional explainability, and structural disentanglement. Unlike traditional GNN explanation methods that typically extract a subgraph from a node's local neighborhood, DISENE introduces a paradigm shift by learning node embeddings where each dimension captures an independent structural feature of the input graph. Additionally, we propose new metrics to evaluate the human interpretability of explanations, analyze the influence of spatial structures and node positions on latent features, and apply post-hoc feature attribution methods to derive task-specific instance-wise explanations.

REFERENCES

Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. *Dynamical processes on complex networks*. Cambridge university press, 2008.

Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. Benchmarking and survey of explanation methods for black box models. *Data Mining and Knowledge Discovery*, 37(5):1719–1778, 2023.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

Sudhanshu Chanpuriya and Cameron Musco. Infinitewalk: Deep network embeddings as laplacian embeddings with a nonlinearity. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1325–1333, 2020.

Sudhanshu Chanpuriya, Ryan Rossi, Anup B Rao, Tung Mai, Nedim Lipka, Zhao Song, and Cameron Musco. Exact representation of sparse networks with symmetric nonnegative embeddings. *Advances in Neural Information Processing Systems*, 36, 2024.

Yu Chen and Mohammed J. Zaki. KATE: K-Competitive Autoencoder for Text. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 85–94, Halifax NS Canada, August 2017. ACM. ISBN 978-1-4503-4887-4. doi: 10.1145/3097983.3098017.

Ayushi Dalmia, Ganesh J, and Manish Gupta. Towards Interpretation of Node Embeddings. In *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18*, pp. 945–952, Lyon, France, 2018. ACM Press. ISBN 978-1-4503-5640-4.

Jingtao Ding, Chang Liu, Yu Zheng, Yunke Zhang, Zihan Yu, Ruikun Li, Hongyi Chen, Jinghua Piao, Huandong Wang, Jiazhen Liu, et al. Artificial intelligence for complex network: Potential, methodology and application. *arXiv preprint arXiv:2402.16887*, 2024.

Keyu Duan, Zirui Liu, Peihao Wang, Wenqing Zheng, Kaixiong Zhou, Tianlong Chen, Xia Hu, and Zhangyang Wang. A comprehensive study on large-scale graph training: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 35:5376–5389, 2022.

Di Fan and Chuanhou Gao. Learning network representations with disentangled graph auto-encoder. *arXiv preprint arXiv:2402.01143*, 2024.

Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

Thorben Funke, Megha Khosla, Mandeep Rathee, and Avishek Anand. Zorro: Valid, sparse, and stable explanations in graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

Antonia Gogoglou, C. Bayan Bruss, and Keegan E. Hines. On the Interpretability and Evaluation of Graph Representation Learning. *NeurIPS workshop on Graph Representation Learning*, 2019.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

William L Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

Darko Hric, Richard K Darst, and Santo Fortunato. Community detection in networks: Structural communities versus ground truth. *Physical Review E*, 90(6):062805, 2014.

Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang. Graphlime: Local interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):6968–6972, 2022.

Zexi Huang, Arlei Silva, and Ambuj Singh. A broader picture of random-walk based graph embedding. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 685–695, 2021.

Maximilian Idahl, Megha Khosla, and Avishek Anand. Finding interpretable concept spaces in node embeddings using knowledge bases. In *Machine Learning and Knowledge Discovery in Databases: International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I*, pp. 229–240. Springer, 2020.

Shima Khoshraftar, Sedigheh Mahdavi, and Aijun An. Centrality-based Interpretability Measures for Graph Embeddings. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10, October 2021. doi: 10.1109/DSAA53316.2021.9564221.

Megha Khosla, Vinay Setty, and Avishek Anand. A comparative study for unsupervised network representation learning. *IEEE Transactions on Knowledge and Data Engineering*, 33(5):1807–1818, 2019.

Dongkwan Kim and Alice Oh. How to find your friendly neighborhood: Graph attention design with self-supervision. In *International Conference on Learning Representations*, 2021.

Konstantin Klemmer, Nathan S Safir, and Daniel B Neill. Positional encoder graph neural networks for geographic data. In *International Conference on Artificial Intelligence and Statistics*, pp. 1379–1389. PMLR, 2023.

Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. Disentangled contrastive learning on graphs. *Advances in Neural Information Processing Systems*, 34:21872–21884, 2021.

Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning. *Advances in Neural Information Processing Systems*, 33:4465–4478, 2020.

Yanbei Liu, Xiao Wang, Shu Wu, and Zhitao Xiao. Independence promoted graph disentangled networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4916–4923, 2020.

Yuheng Lu, Jinpeng Chen, ChuXiong Sun, and Jie Hu. Graph inference representation: Learning graph positional embeddings with anchor path encoding. *arXiv preprint arXiv:2105.03821*, 2021.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

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

Lucie Charlotte Magister, Dmitry Kazhdan, Vikash Singh, and Pietro Liò. Gcexplainer: Human-in-the-loop concept-based explanations for graph neural networks. *arXiv preprint arXiv:2107.11889*, 2021.

Lucie Charlotte Magister, Pietro Barbiero, Dmitry Kazhdan, Federico Siciliano, Gabriele Ciravegna, Fabrizio Silvestri, Mateja Jamnik, and Pietro Lio. Encoding concepts in graph neural networks. *arXiv preprint arXiv:2207.13586*, 2022.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

Simone Piaggesi, Megha Khosla, André Panisson, and Avishek Anand. Dine: Dimensional interpretability of node embeddings. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10772–10781, 2019.

Thibault Prouteau, Nicolas Dugué, Nathalie Camelin, and Sylvain Meignier. Are embedding spaces interpretable? results of an intrusion detection evaluation on a large french corpus. In *LREC 2022*, 2022.

Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 459–467, 2018.

Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. GEMSEC: graph embedding with self clustering. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 65–72, Vancouver British Columbia Canada, August 2019. ACM. ISBN 978-1-4503-6868-1.

Guillaume Salha, Romain Hennequin, and Michalis Vazirgiannis. Simple and effective graph autoencoders with one-hop linear models. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*, pp. 319–334. Springer, 2021.

Benjamin Sanchez-Lengeling, Jennifer Wei, Brian Lee, Emily Reif, Peter Wang, Wesley Qian, Kevin McCloskey, Lucy Colwell, and Alexander Wiltschko. Evaluating attribution for graph neural networks. *Advances in neural information processing systems*, 33:5898–5910, 2020.

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.

Lütfi Kerem Şenel, Ihsan Utlu, Veysel Yücesoy, Aykut Koc, and Tolga Cukur. Semantic structure and interpretability of word embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1769–1779, 2018.

Zohair Shafi, Ayan Chatterjee, and Tina Eliassi-Rad. Generating human understandable explanations for node embeddings. *arXiv preprint arXiv:2406.07642*, 2024.

Dougal Shakespeare and Camille Roth. Interpreting node embedding distances through n-order proximity neighbourhoods. In *International Conference on Complex Networks*, pp. 179–193. Springer, 2024.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. SPINE: SParse Interpretable Neural Embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. ISSN 2374-3468. Number: 1.

Anton Tsitsulin, Marina Munkhoeva, Davide Mottin, Panagiotis Karras, Ivan Oseledets, and Emmanuel Müller. Frede: anytime graph embeddings. *Proceedings of the VLDB Endowment*, 14(6): 1102–1110, 2021.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

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

Xin Wang, Hong Chen, Si'ao Tang, Zihao Wu, and Wenwu Zhu. Disentangled representation learning. *arXiv preprint arXiv:2211.11695*, 2022.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pp. 6861–6871. PMLR, 2019.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Han Xuanyuan, Pietro Barbiero, Dobrik Georgiev, Lucie Charlotte Magister, and Pietro Liò. Global concept-based interpretability for graph neural networks via neuron analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10675–10683, 2023.

Jaewon Yang and Jure Leskovec. Community-affiliation graph model for overlapping network community detection. In *2012 IEEE 12th international conference on data mining*, pp. 1170–1175. IEEE, 2012.

Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 587–596, 2013.

Yiding Yang, Zunlei Feng, Mingli Song, and Xinchao Wang. Factorizable graph convolutional networks. *Advances in Neural Information Processing Systems*, 33:20286–20296, 2020.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *International conference on machine learning*, pp. 7134–7143. PMLR, 2019.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *International Conference on Machine Learning*, pp. 12241–12252. PMLR, 2021.

Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(5): 5782–5799, 2022.

Tianqi Zhao, Thi Ngan Dong, Alan Hanjalic, and Megha Khosla. Multi-label node classification on graph-structured data. *Transactions on Machine Learning Research*.

A APPENDIX / SUPPLEMENTAL MATERIAL

A.1 DATASET STATISTICS

Table A1: Summary statistics of graph-structured data. In empirical data, we restrict our analysis to the largest connected component of any graph.

	CORA	WIKI	FB	PPI	RING-CL	SBM	BA-CL	ER-CL	TR-CL	TR-GR
# nodes	2,485	2,357	4,039	3,480	320	320	640	640	831	799
# edges	5,069	11,592	88,234	53,377	1,619	1,957	3,138	4,196	2,081	972
# clusters/motifs	28	18	16	9	32	32	32	32	32	32
density	0.002	0.004	0.011	0.009	0.032	0.038	0.015	0.021	0.006	0.003
clust. coeff.	0.238	0.383	0.606	0.173	0.807	0.561	0.486	0.456	0.360	0.002

Table A2: Summary statistics of graph biological data used for multi-label node classification.

	PPI	PCG	HUMLOC	EUKLOC
# nodes	3,480	3,177	2,552	2,969
# edges	53,377	37,314	15,971	11,130
# labels	121	15	14	22
density	0.009	0.007	0.005	0.003
clust. coeff.	0.173	0.346	0.132	0.150

A.2 TRAINING SETTINGS

- For DEEPWALK (Perozzi et al., 2014), we train NODE2VEC[5] algorithm for 5 epochs with the following parameters: `p`$= 1$, `q`$= 1$, `walk_length`$= 20$, `num_walks`$= 10$, `window_size`$= 5$.
- For INFWALK[6] (Chanpuriya & Musco, 2020), a matrix factorization-based method linked to DEEPWALK and spectral graph embeddings, we set the same value `window_size`$= 5$ used for DEEPWALK.
- In GRAPHAE (Salha et al., 2021), we optimize a 1-layer GCN encoder with a random-walk loss setting analogous to DEEPWALK. The model is trained for 50 iterations using Adam optimizer and learning rate of 0.01.
- in GRAPHSAGE[7] (Kim & Oh, 2021), we optimize a 2-layer SAGE encoder with `mean` aggregation and with a random-walk loss setting analogous to DEEPWALK. The model is trained for 50 iterations using Adam optimizer, learning rate of 0.01.
- DINE[8] (Piaggesi et al., 2024), autoencoder-based post-processing process trained for 2000 iterations, and learning rate of 0.1. Input embeddings are from DEEPWALK and GAE methods, tuning the input embedding size in the list $[8, 16, 32, 64, 128, 256, 512]$.
- DISE-FCAE and DISE-GAE trained for 50 iterations using Adam optimizer and learning rate of 0.01. Random walk sampling follows the same setting as DEEPWALK, GRAPHAE and GRAPHSAGE.

[5] https://github.com/eliorc/node2vec
[6] https://github.com/schariya/infwalk
[7] https://github.com/pyg-team/pytorch_geometric/blob/master/examples/graph_sage_unsup.py
[8] https://www.github.com/simonepiaggesi/dine

A.3 ALGORITHM COMPLEXITY

Space and time complexity of DISENE can be analyzed by looking at the pseudo-code in Algorithm A1. Part of the complexity depends on the complexity of the encoder. Here, we assume GCN as encoding functions, with its own set of learnable parameters Θ. But, in the experiments, we have also tested fully-connected encoders.

Algorithm A1: DISENE($\mathcal{G}, \mathbf{A}, K, T, L, \lambda_{\text{ent}}$)

Input : Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Adjaceny matrix $\mathbf{A} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|}$
Embedding size K, Context window T,
Walks length L, Regularization λ_{ent}
Output : Embedding matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}|\times K}$

Init. encoder network $Enc_\Theta(*)$;
Init. identity matrix features $\mathbf{X}$;
while ***not converged*** **do**
  Encoding step: $\mathbf{H} \leftarrow \rho(\mathbf{W}^\top Enc_\Theta(\mathbf{A}, \mathbf{X}))$;
  Sample batch of nodes: $\mathcal{B} \leftarrow Sample(\mathcal{V})$;
  Init. random walks $\mathcal{W} \leftarrow \emptyset$;
  foreach $v \in \mathcal{B}$ **do**
    Sample random walk:
    $\mathcal{W} \leftarrow \mathcal{W} \cup RandomWalk(\mathbf{A}, v, L)$;
  Random-walk loss: $\mathcal{L}_{\text{rw}}(\mathbf{H}, \mathcal{W}, T)$;
  foreach $d \in \{1 \dots K\}$ **do**
    Aggregate rows of $\mathbf{H}$: $\mathbf{f}_d \leftarrow \sum_v \mathbf{H}_{vd}$;
    Compute 1-norm: $|\mathbf{f}|_1 \leftarrow \sum_{v,d} \mathbf{H}_{vd}$;
  Node affiliation matrix: $\mathbf{F} \leftarrow \mathbf{H} \odot \mathbf{f}$;
  Disentanglement loss $\mathcal{L}_{\text{dis}}(\mathbf{F})$
  Regularization loss $\mathcal{L}_{\text{ent}}(\mathbf{F})$
  Total loss: $\mathcal{L} \leftarrow \mathcal{L}_{\text{rw}} + \mathcal{L}_{\text{dis}} + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}}$;
  Backpropagate and update $\Theta, \mathbf{W}$;
return $\mathbf{H}$;

Our method consists of four main steps:

- Encoding step generates the node embeddings $\mathbf{H}$ and has the same per-layer time/space complexity of standard GCNs (Duan et al., 2022), i.e. $\mathcal{O}(||\mathbf{A}||_0 K + |\mathcal{V}|K^2)$ and $\mathcal{O}(|\mathcal{V}|K)$ respectively.
- Random walk sampling and loss calculation has time/space complexity $\mathcal{O}(|\mathcal{V}|KTL)$ and $\mathcal{O}(|\mathcal{V}|L)$ respectively (Rozemberczki et al., 2019), where T is the context window size and L is the random-walk length (we sample 1 random walk per node, fixing as well the number of negative samples to 1 for each positive sample). $RandomWalk$ function sample a first-order random walk starting from source node v of length L.
- Node affiliation matrix involves computing the entries $\mathbf{F}_{ud} = \sum_{v\in\mathcal{V}_d} \phi_d(u, v; \mathbf{h})$ as $\mathbf{F}_{ud} = \sum_v \mathbf{H}_{ud}\mathbf{H}_{vd} = \mathbf{H}_{ud}\mathbf{f}_d$, i.e. by multiplying node embedding entries $\mathbf{H}_{ud}$ with quantities $\mathbf{f}_d = \sum_v \mathbf{H}_{vd}$. This step involves $\mathcal{O}(|\mathcal{V}|K)$ operations for computing and storing matrix $\mathbf{F}$.
- Disentanglement and regularization losses involve respectively $\mathcal{O}(|\mathcal{V}|K^2)$ and $\mathcal{O}(K)$ operations for cosine similarity (matrix products) and entropy (vector sum).

Overall, given that $||\mathbf{A}||_0$ is $2|\mathcal{E}|$, DISENE results in $\mathcal{O}(|\mathcal{E}|K+|\mathcal{V}|K^2+|\mathcal{V}|KTL)$ runtime complexity and $\mathcal{O}(|\mathcal{V}|K + |\mathcal{V}|L)$ space complexity, which are in line with established node embedding methods (see, for instance, Table 1 in Tsitsulin et al. (2021) for an exhaustive summary).

A.4 DOWNSTREAM TASKS RESULTS

We tested link prediction for the datasets reported in the main paper. For node classification, we tested PPI and other benchmark biological datasets in multi-label setting (Zhao et al.): the PCG dataset for the protein phenotype prediction, the HUMLOC, and EUKLOC datasets for the human and eukaryote protein subcellular location prediction tasks, respectively. Characteristics of additional biological datasets are reported in Table A2. We concatenated node attributes to node embeddings to get an enriched set of predictors that, given our method extract interpretable features, can be used in combination with feature-based explainers (e.g., SHAP) for building fully transparent prediction pipelines. In Figure A1 we report AUC-PR scores for link prediction and node classification in real-world graph data. Generally, scores increase with the number of latent embedding dimensions. Tables A3 and A4 show the maximum scores for link prediction and node classification, demonstrating that our approach can consistently achieve reasonable performances within the expected range of the performance-interpretability trade-off.

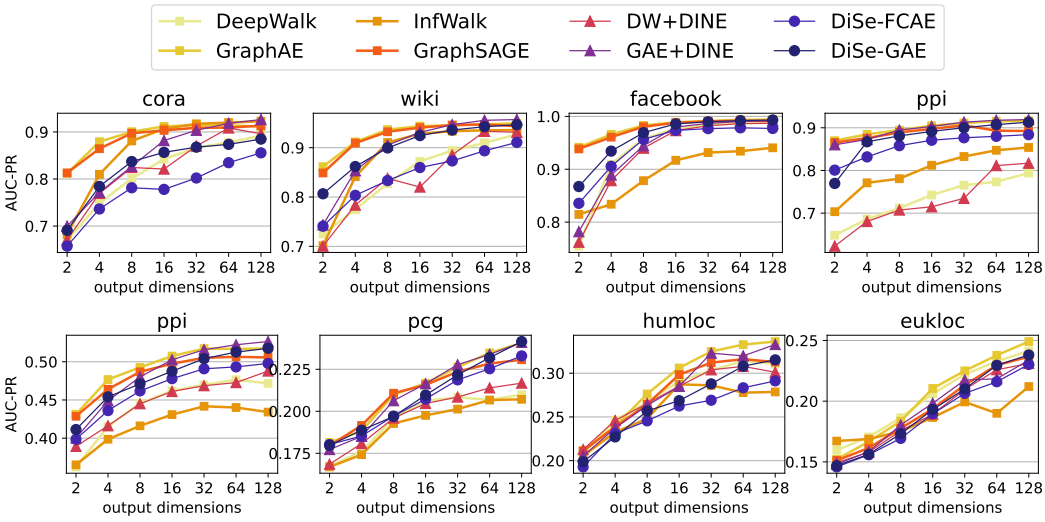

Figure A1: Downstream tasks results on real-world datasets (link prediction on the top panel, multi-label node classification on the bottom panel) with varying feature dimensions size.

Table A3: Link prediction results (AUC-PR) on real-world datasets. Best scores are in bold, while scores with a relative performance loss of no more than 2% respect to the best score are underlined.

	CORA	WIKI	FB	PPI
DEEPWALK	.892±.005	.927±.002	.990±.001	.794±.002
GRAPHAE	.911±.003	.950±.001	**.994**±.001	.916±.001
INFWALK	.923±.003	.936±.002	.941±.006	.854±.003
GRAPHSAGE	.913±.005	.944±.002	.991±.001	.892±.003
DW+DINE	.896±.004	.931±.003	.987±.001	.817±.004
GAE+DINE	**.926**±.001	**.957**±.003	.992±.002	**.919**±.002
DISE-FCAE	.856±.007	.911±.004	.977±.001	.884±.002
DISE-GAE	.885±.002	.947±.002	.993±.006	.913±.001

Table A4: Node classification results (AUC-PR) on real-world datasets. Best scores are in bold, while scores with a relative performance loss of no more than 5% respect to the best score are underlined.

	PPI	PCG	HUMLOC	EUKLOC
DEEPWALK	.476±.003	.210±.001	.314±.012	.241±.010
GRAPHAE	.517±.003	.241±.001	**.336**±.004	**.249**±.005
INFWALK	.442±.001	.207±.002	.287±.004	.212±.003
GRAPHSAGE	.506±.001	.231±.002	.316±.004	.237±.011
DW+DINE	.488±.002	.217±.001	.308±.004	.231±.008
GAE+DINE	**.526**±.001	.241±.001	.333±.006	.234±.008
DISE-FCAE	.498±.001	.233±.003	.291±.006	.230±.006
DISE-GAE	.518±.001	**.242**±.004	.315±.003	.238±.006

A.5 GLOBAL EXPLANATIONS RESULTS FOR SYNTHETIC DATASETS

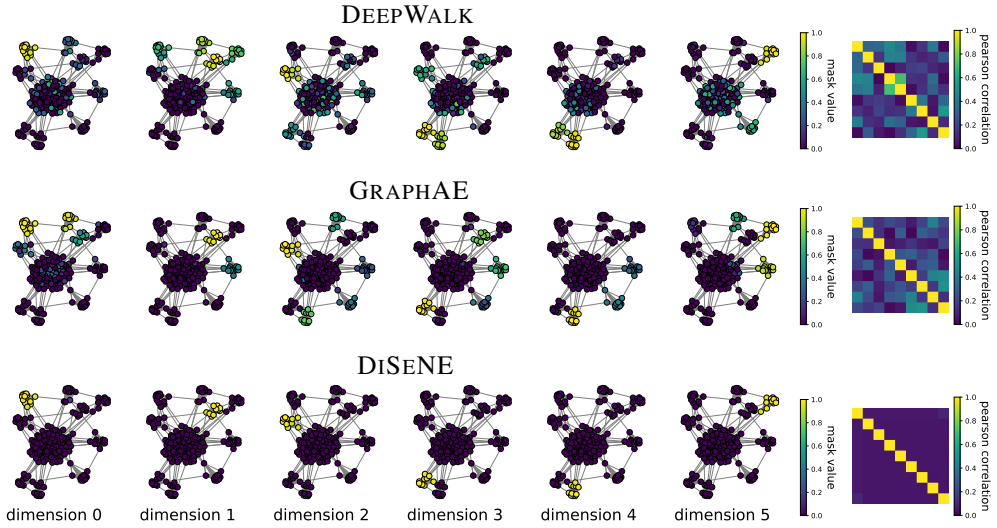

Figure A2: Subgraph-level global explanations for a representative subset of embedding dimensions, along with corresponding pairwise feature correlation plots, on synthetic dataset BA-CLIQUES.

A.5.1 VISUALIZATION OF SUBGRAPH EXPLANATIONS

Algorithm A2:
UNSUPEDGESUBGRAPH$(\mathcal{G}, \mathbf{A}, \mathbf{Z}, d)$

Input : Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Embedding function $\mathbf{z} : \mathcal{V} \to \mathbb{R}^K$
Dimension to explain $d \in \{1 \dots K\}$
Output : Graph mask $\mathbf{M}^{(d)} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$

Init. graph mask: $\mathbf{M}^{(d)} \leftarrow 0^{|\mathcal{V}| \times |\mathcal{V}|}$;
Compute background average attribution: $\zeta_d = \frac{1}{|\mathcal{E}|} \sum_{(u,v) \in \mathcal{E}} z_d(u) z_d(v)$;
for $(u, v) \in \mathcal{E}$ **do**
  Compute edge attribution: $\phi_d(u, v; \mathbf{z}) = z_d(u) z_d(v) - \zeta_d$;
  Add explanation: $\mathbf{M}^{(d)}_{uv} \leftarrow \max\{0, \phi_d(u, v; \mathbf{z})\}$;
return $\mathbf{M}^{(d)}$;

In Figure A2 we show subgraph-level global explanations on synthetic dataset BA-CLIQUES. Subgraphs are generated for each feature dimension using the procedure described in Section 3.1 (summarized on the left in Algorithm A2) and are based on various unsupervised embedding methods. The explanatory subgraphs demonstrate that our method effectively aligns embedding dimensions with meaningful, non-random functional components of the graph. In contrast, standard methods such as DEEPWALK and GRAPHAE struggle to isolate individual structural units within dimensions. Instead, their embeddings often associate dimensions with groups of cliques or subgraphs that include elements from the random Barabási-Albert scaffold. Additionally, the visualization on the right shows the correlation between latent features, further underscoring that the alignment between embedding dimensions and graph structure is closely tied to the ability to disentangle feature correlation through non-collinearity.

A.5.2 EXTENSIVE RESULTS FOR SYNTHETIC DATASETS

In Figure A3 we plot results for Comprehensibility and Sparsity, on the top and the bottom respectively, on synthetic datasets. Generally, DISE-FCAE outperforms DISE-GAE and the other competitors in all the datasets. In Figure A4 we plot results for Overlap Consistency and Positional Coherence, on the top and the bottom respectively, on synthetic datasets. For the overlap metric, DISE-FCAE and DISE-GAE consistently outperform the competitors, especially with more than 8 dimensions where they achieve almost perfect overlap. For the positional metric, the competitors GAE+DINE and DW+DINE slightly outperform DISE methods, especially in large dimensions, while DEEPWALK also show good results.

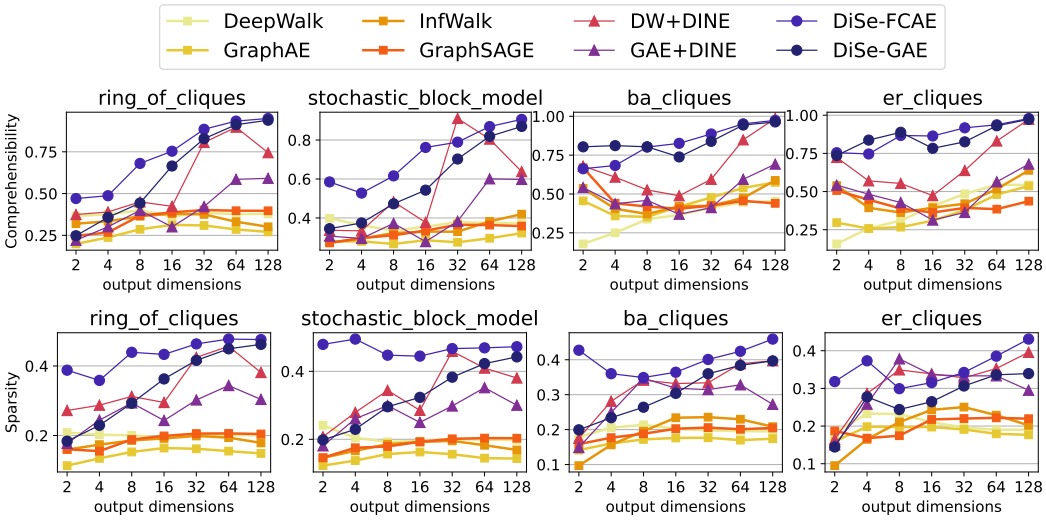

Figure A3: Comprehensibility and sparsity results on synthetic datasets with varying feature dimensions size.

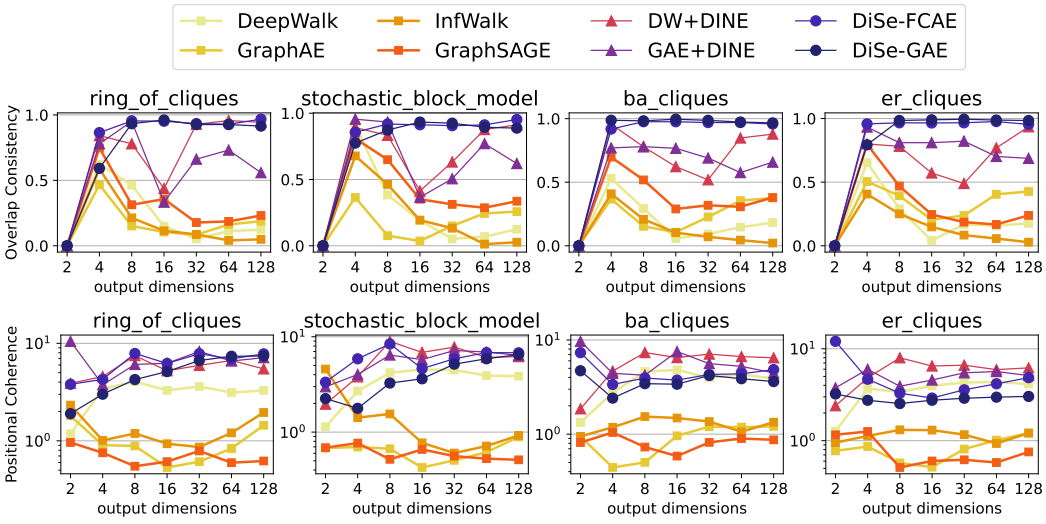

Figure A4: Overlap consistency and positional coherence results on synthetic datasets with varying feature dimensions size.

A.6 LOCAL EXPLANATIONS RESULTS FOR SYNTHETIC DATASETS

A.6.1 EXPLANATION TASK DESCRIPTION IN DETAILS

Local explanations for node embeddings are extracted by using post-hoc feature importance method SHAP. For a given embedding model $\mathbf{h} : \mathcal{V} \leftarrow \mathbb{R}^K$ we train a downstream classifier, e.g., in node classification task or link prediction. For simplicity, here we write the case when the classifier is a (binary) linear model, but it can be any arbitrary complex model. It is anyway reasonable to assume that node embeddings come from a deep graph model and downstream classifier is a simple 1-layer neural network on top of the embedding layers.

Algorithm A3:
NODECLASSSUBGRAPH$(\mathcal{G}, \mathbf{A}, \mathbf{\Psi}, j)$

Input : Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Feature-base explanation matrix $\mathbf{\Psi} \in \mathbb{R}^{|\mathcal{V}| \times K}$
Dimension to explain $j \in \{1 \dots K\}$
Output : Node mask $\mathbf{B}^{(j)} \in \mathbb{R}^{|\mathcal{V}|}$

Init. node mask: $\mathbf{B}^{(j)} \leftarrow 0^{|\mathcal{V}|}$;
for $v \in \mathcal{V}$ **do**
  Add explanation:
  $\mathbf{B}_u^{(j)} \leftarrow \max\{0, \mathbf{\Psi}_{vj}\}$;
return $\mathbf{B}^{(j)}$;

$$\textit{(node classification)} \quad b(v) = \sigma(\sum_{j=1}^{K} \beta_j h_j(v) + \beta_0)$$

$$\textit{(link classification)} \quad b(u, v) = \sigma(\sum_{j=1}^{K} \beta_j h_j(u, v) + \beta_0)$$

Given a vector representation of a graph instance (e.g., a node embedding $\mathbf{h}(v)$ or an edge embedding $\mathbf{h}(u, v)$), and the corresponding prediction from classifier b, we compute feature importance with SHAP $\{\mathbf{\Psi}_{v,j}^{(\mathcal{V})}\}_{j=1\dots K}$ or $\{\mathbf{\Psi}_{(uv),j}^{(\mathcal{E})}\}_{j=1\dots K}$ and the corresponding task-based graph masks (we illustrate the pesudo-code for node classification masks in Algorithm A3):

$$\mathbf{B}^{(j)}(\mathbf{\Psi}^{(\mathcal{V})}) \in \mathbb{R}^{|\mathcal{V}|}; \quad B_v^{(j)} = \max\{0, \mathbf{\Psi}_{v,j}^{(\mathcal{V})}\}$$

$$\mathbf{B}^{(j)}(\mathbf{\Psi}^{(\mathcal{E})}) \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}; \quad B_{uv}^{(j)} = \max\{0, \mathbf{\Psi}_{(uv),j}^{(\mathcal{E})}\}$$

It is valuable to remark that, training with logistic regression and applying SHAP, the resulting importance scores are simply the coefficients of the regression (Lundberg & Lee, 2017) $\mathbf{\Psi}_{x,j} = \beta_j(h_j(x) - E[h_j])$. Thus, combining this methodology to interpretable graph features of DISENE, we obtain a fully transparent node/edge classification pipeline for graph data.

A.6.2 COMPARISON OF EMBEDDING METHODS

In Figure A5 we plot results for the plausibility metric on link prediction and node classification, on the top and the bottom respectively, while comparing different unsupervised methods that output node embeddings. Plausibility seems to benefit larger dimension values for DISE methods and DW+DINE for link prediction. Figure A6 shows the corresponding downstream task accuracy results.

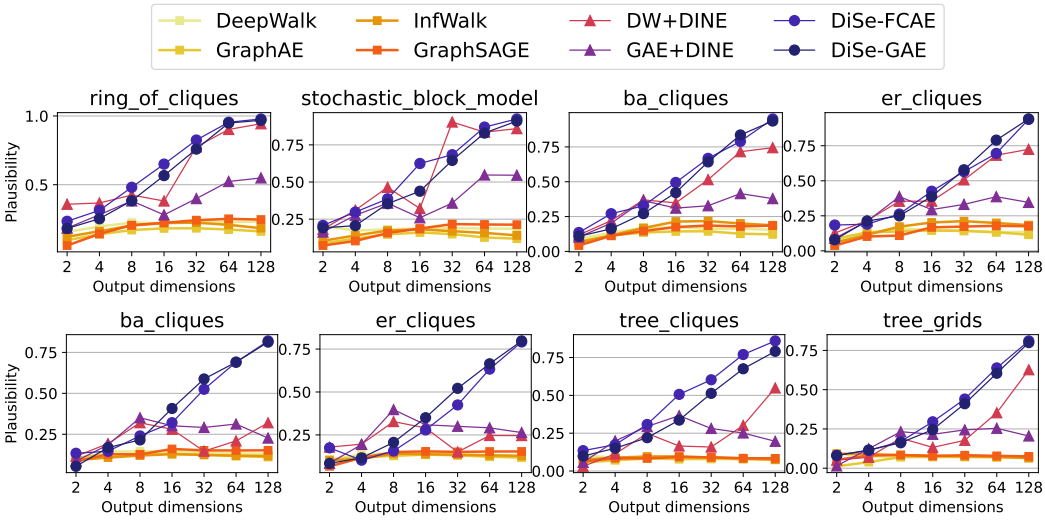

Figure A5: Plausibility results on synthetic datasets (link prediction on the top panel, binary node classification on the bottom panel) with varying feature dimensions size.

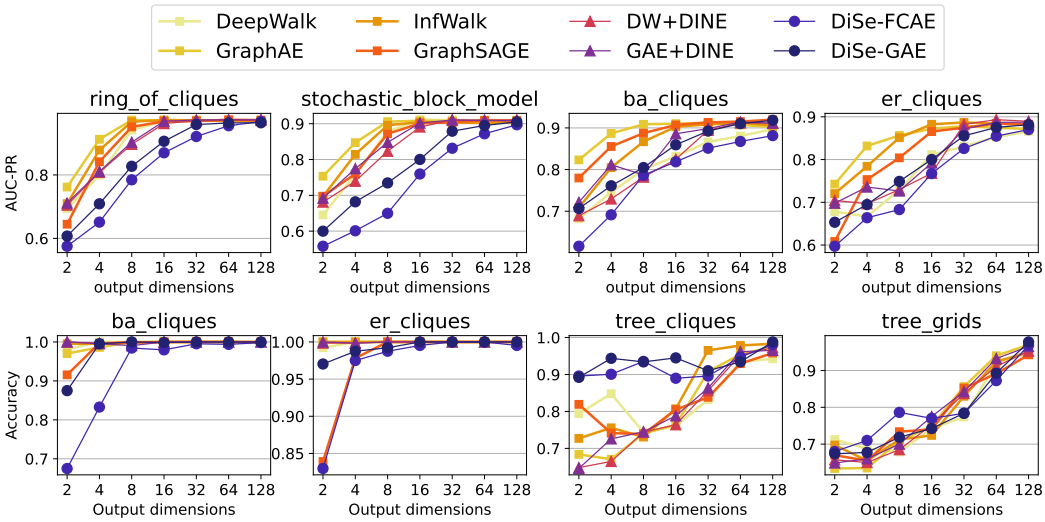

Figure A6: Downstream tasks results on synthetic datasets (link prediction on the top panel, binary node classification on the bottom panel) with varying feature dimensions size.

A.6.3 COMPARISON WITH LOCAL EXPLAINER METHODS FOR GRAPHS

In this section, we report Plausibility results of local explanations for node classification tasks on synthetic data, in comparison with standard local graph explainers: GNNEXPLAINER (Ying et al., 2019) and PGEXPLAINER (Luo et al., 2020). These methods present local explanation in the form of node and/or edge importance, whereas in our method –combined with feature-based explainer– the explanation format is a vector of feature importance, associated with a subgraph for each feature. For a fair comparison, we consider as the explanation presented by our method the subgraph associated to the most important embedding feature (according to the logistic classifier). Recalling Eq. (9) in the main paper for Plausibility, this approach is equivalent to choosing as f the function:

$$f(\Psi_j) = \begin{cases} 1, & \text{if } j = \underset{d \in \{1 \dots K\}}{\operatorname{argmax}} \Psi_d \\ 0, & \text{otherwise} \end{cases}$$

We compare Plausibility from computed node masks for the test node instances. GNNEXPLAINER[9] is trained for 30 epochs for each test node, while PGEXPLAINER[10] is trained for 5 epochs on trained nodes before being applied on test nodes. Moreover, since PGEXPLAINER is based on edge masks, we derive node masks for that model with the average mask value from incident edges. Plausibility results are computed over test nodes with correct predicted label, because the explanations extracted from wrong predictions are not reliable for analyzing local model decisions. Graph explainers are applied on the output of the following 2-layer GNNs method trained on node classification: GCN (Wu et al., 2019), GRAPHSAGE (Hamilton et al., 2017), and GATV2 (Brody et al., 2022). All the graph models (not the explainers) are tuned by searching the best embedding size from the list $[2, 4, 8, 16, 32, 64, 128]$, as the input to the classification layer.

Table A5: Plausibility results for synthetic datasets with best scores in bold, second best underlined.

Method		Node Classification			
		BA-CLIQUES	ER-CLIQUES	TREE-CLIQUES	TREE-GRIDS
GNNEXPL	GCN	.729±.004	.638±.005	.846±.004	.809±.003
	GRAPHSAGE	.703±.006	.611±.005	.829±.005	.810±.002
	GATV2	.707±.004	.633±.002	.832±.006	.808±.004
PGEXPL	GCN	<u>.895</u>±.005	**.923**±.004	.863±.009	.573±.003
	GRAPHSAGE	.581±.038	.704±.032	.374±.006	.695±.025
	GATV2	.596±.016	.724±.002	.422±.074	.721±.030
DISE-FCAE		**.919**±.001	<u>.881</u>±.006	**.926**±.001	<u>.889</u>±.006
DISE-GAE		.875±.009	.872±.008	<u>.871</u>±.005	**.898**±.001

We test explanation methods with synthetic datasets BA-CLIQUES, ERCLIQUES, TREE-CLIQUES and TREE-GRIDS (Ying et al., 2019). From Table A5, we observe GNNEXPLAINER has uniform results across different input GNN models, instead PGEXPLAINER performs best with GCN. DISE-FCAE and DISE-GAE outperforms the competitors in most of the cases, except with GCN+PGEXPLAINER in ER-CLIQUES.

Figure A7 present examples of local explanations for node classification tasks on the small-sized synthetic datasets BA-CLIQUES and TREE-GRIDS, using different methods. The experimental settings are consistent with those described above. Notably, DISENE demonstrates a strong ability to produce meaningful and interpretable node masks, effectively competing with state-of-the-art GNN explanation methods.

[9] https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.explain.algorithm.GNNExplainer.html#torch_geometric.explain.algorithm.GNNExplainer

[10] https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.explain.algorithm.PGExplainer.html#torch_geometric.explain.algorithm.PGExplainer

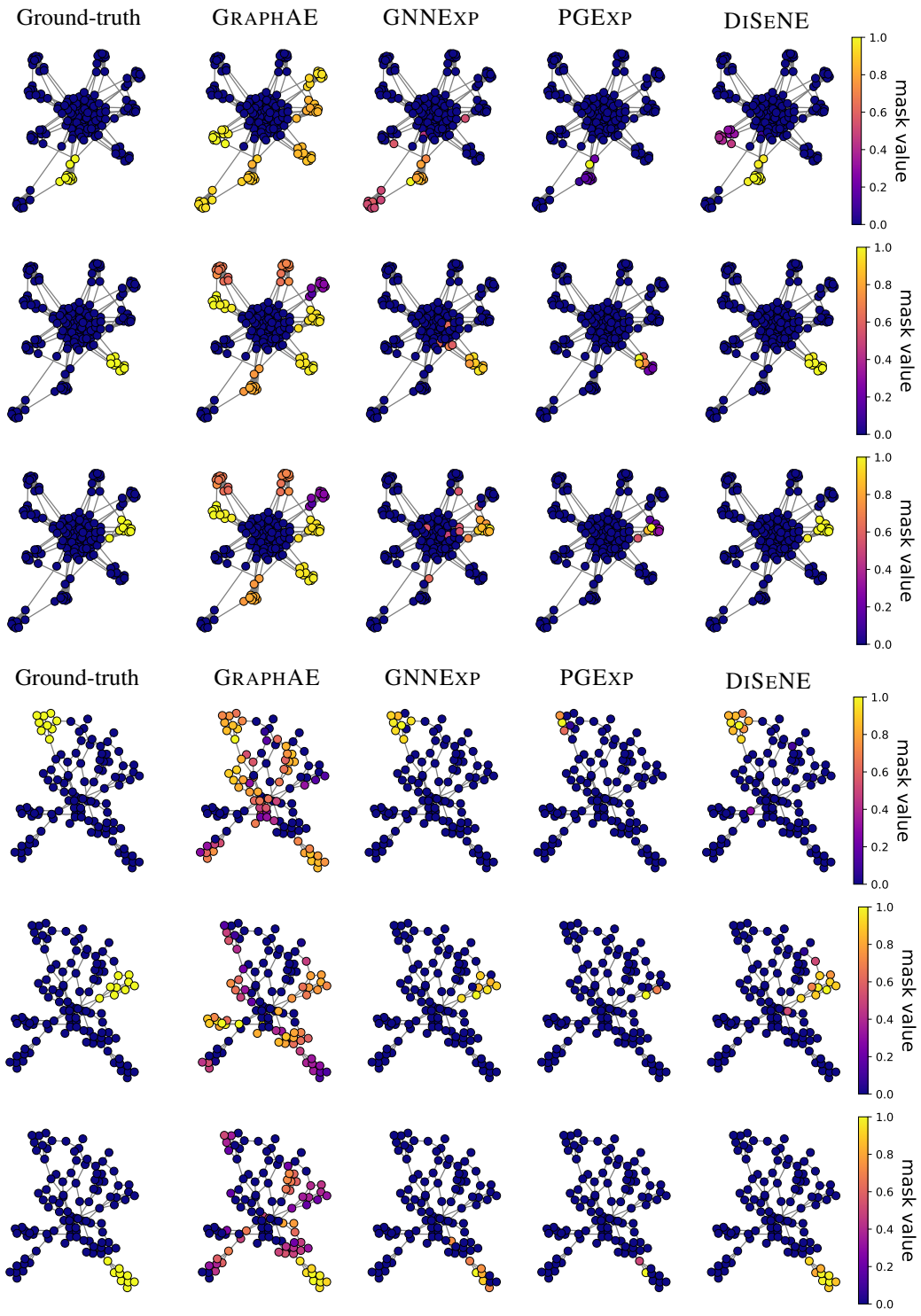

Figure A7: Subgraph local explanations for node classification in BA-CLIQUES (top) and TREE-GRIDS (bottom). On the leftmost column, we highlight the local ground-truth structures for the instance nodes depicted in the illustrations. On the other columns, we display the explanation subgraphs generated by each method, with nodes color-coded according to the respective explanation masks. For GRAPHAE and DISENE, the visualized subgraphs represent the most relevant structures extracted with Algorithm A3 and determined by feature importance attribution from the logistic regression node classifier. For GNNEXPLAINER and PGEXPLAINER, the node masks correspond to the algorithm's output in explaining a 2-layer GCN (Wu et al., 2019) trained on node classification.