GNUMAP: A Parameter-Light Approach to Unsupervised Dimensionality Reduction via Graph Neural Networks

Jihee You
jiheeyou@uchicago.edu
University of Chicago
Chicago, Illinois, USA

So Won Jeong
sowonjeong@uchicago.edu
University of Chicago
Chicago, Illinois, USA

Claire Donnat
cdonnat@uchicago.edu
University of Chicago
Chicago, Illinois, USA

ABSTRACT

With the proliferation of Graph Neural Network (GNN) methods stemming from self-supervised and contrastive learning, unsupervised node representation learning is rapidly gaining traction across various fields. It has particularly gained popularity in the analysis of complex systems, such as in biology or molecular dynamics, where it is often used as a dimensionality reduction tool. However, there remains a significant gap in understanding the quality of the low-dimensional node representations these methods produce, particularly in scenarios that go beyond the scope of commonly used, well-curated academic datasets for node classification. To address this gap, we propose here the first comprehensive benchmarking of various unsupervised node embedding techniques tailored for dimensionality reduction, encompassing a range of manifold learning tasks, along with various performance metrics. *We emphasize the sensitivity of current methods to hyperparameter choices — highlighting a fundamental issue as to their applicability in real-world settings where there is no established methodology for rigorous hyperparameter selection.* Addressing this issue, we introduce GNUMAP, a more robust method for unsupervised node representation learning that merges the traditional UMAP approach with the expressivity of the GNN framework. We show that GNUMAP consistently outperforms existing state-of-the-art GNN embedding methods in a variety of contexts, including synthetic geometric datasets, citation networks, and real-world biomedical data — making it a simple but reliable dimensionality reduction tool. We hope this paper highlights the critical importance of carefully selecting the node representation learning technique and paves the way for a more robust assessment of the potential use of GNNs as a dimensionality reduction tool.

KEYWORDS

Unsupervised embedding; simulated data; biomedical application

ACM Reference Format:
Jihee You, So Won Jeong, and Claire Donnat. 2024. GNUMAP: A Parameter-Light Approach to Unsupervised Dimensionality Reduction via Graph Neural Networks. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Consider the following biological problem: given a set of gene expressions at various locations within a tissue sample (for instance, a slice of mouse brain tissue), how can we aggregate information to discover spatial domains with coherent gene expression patterns? Recent methods, including [14] from which this example is adapted, have turned towards using a graph-based approach — and subsequently, Graph Neural Networks (GNNs) — to resolve this conundrum. In this setting, the data is first represented as a graph $\mathcal{G}$ on n nodes corresponding here to the various spatial locations within the sample, each endowed with a feature vector $X_v \in \mathbb{R}^d$ (the gene expression data). Under this new formalism, Graph Neural Networks (GNNs) [16, 26] come as a natural tool for visualization and the subsequent discovery of new patterns in the data. Through the use of recursive neighborhood "convolutions", GNNs allow indeed the creation of rich node representations that capture topological information, feature data and essential neighborhood properties, integrating this information in a Euclidean vector representation that is amenable to any downstream machine learning task. Heralded as the breakthrough for machine learning on graphs that would allow the same "AI renaissance" [3] that standard neural networks have brought to Computer Vision and Natural Language Processing, GNNs have been suggested as a panacea for a wide number of tasks across disciplines, including molecular design [10, 27], traffic prediction [6, 8], biological networks [18, 19, 38] or recommender systems.

Unsupervised Learning on Graph Data. While the early rise in Graph Neural Network (GNN) advancements predominantly emphasized supervised learning, there has been an increasing interest in developing unsupervised GNN methods for learning node representations. This shift is driven by the scarcity of labeled data in numerous real-world situations, such as in the example described above, coupled with an increasing demand for techniques that facilitate exploratory data analysis and dimensionality reduction for graph data. Yet, there seems to have been a concurrent development of unsupervised learning approaches in the applications community [14, 15, 17, 24, 36] and in the methods community [13, 29, 31, 35, 37]. This parallel development may underscore a disconnect between the approaches devised by the methods community and the practical reality of real-world data. Specifically, the deployment of state-of-the-art approaches evaluated on academic benchmarks seems to be impeded by two significant challenges: *(a) Ease of deployment*, and *(b) Trustworthiness of the learned representations.*

(a) Ease of deployement. From a methodological standpoint, a growing number of recent approaches are leaning toward the adoption of self-supervised contrastive learning frameworks to effectively represent nodes [13, 29, 31, 35, 37]. In this setting, the trick

KDD, August 25–29, 2024, Barcelona, Spain

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

usually consists of perturbing the input data (e.g., by masking features with probability p_f, dropping edges with probability p_e) to create two modified versions of the original data. Each of these perturbed versions of the data is then passed through a Graph Neural Network, which is trained by learning to recognize which pairs of node embeddings — stemming from each of the two perturbed graphs — represent the same node in the original dataset. Without getting into the specifics of the architecture, examples of such training objectives include the loss proposed by [37] for their method GRACE, $\mathcal{L} = \frac{1}{2N}\left(\sum_{i=1}^{N}\left[\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) + \ell(\boldsymbol{v}_i, \boldsymbol{u}_i)\right]\right)$, where

$$\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) = \log\left(\frac{e^{s(\boldsymbol{u}_i,\boldsymbol{v}_i)/\tau}}{\sum\limits_{k=1}^{N}\left[\mathbf{1}_{[\mathrm{k}\neq\mathrm{i}]}e^{s(\boldsymbol{u}_i,\boldsymbol{u}_k)/\tau} + e^{s(\boldsymbol{u}_i,\boldsymbol{v}_k)/\tau}\right]}\right), s : \mathbb{R}^F \times \mathbb{R}^F \rightarrow$$

$\mathbb{R}$ is the cosine similarity function, and $\boldsymbol{u}_i, \boldsymbol{v}_i \in \mathbb{R}^F$ are the node representation for node i stemming from the two perturbed versions of the graph. Finally, $\tau > 0$ is a temperature parameter which has to be tuned. Another example is the loss use in CCA-SSG [35], $\mathcal{L} = \sum_{i=1}^{n}\|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 + \lambda\left(\left\|\boldsymbol{U}^T\boldsymbol{U} - \boldsymbol{I}\right\|_F^2 + \left\|\boldsymbol{V}^T\boldsymbol{V} - \boldsymbol{I}\right\|_F^2\right)$ where $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{N\times F}$ are the node representation matrix of two views, $\lambda > 0$ is a hyperparameter, and $\|\cdot\|_F$ denotes the Frobenius norm. While these self-supervised learning losses have achieved state-of-the-art performance in a number of academic benchmarks, few of these methods have yet been deployed in applied settings. One hypothesis explaining this disparity is the heavy reliance of these "state-of-the-art" approaches on the correct choice of hyperparameters (λ, τ, p_f and p_e). We exemplify this phenomenon in Figure 1, where we propose to use a simple 2-layer GCN combined with the CCA-SSG [35] learning framework to learn a 2D visualization of the nodes. We then assess the quality of the learned embeddings by learning a support vector classifier to classify the nodes, and evaluate the performance of the classification on held-out data. We note a substantial variation in embedding quality as the edge drop rate p_e, the feature mask rate p_m and the regularization parameter λ vary. This highlights the importance of selecting the "correct" hyperparameters: a wrong choice of hyperparameters could push the method to significantly underperform or to learn uninformative embeddings (see Figure 1 left). However, in the unsupervised context, there is no established cross-validation technique for GNNs or other principled technique for parameter selection at large. This significantly complicates the practical application of these methods in settings where there are no labels to evaluate the method on for hyperparameter selection.

(b) Trustworthiness. Moreover, despite the increasing adoption of graph neural networks (GNNs), there is a notable gap in research focusing on the evaluation of the quality of GNN node representations: How effectively do these embeddings capture the structural details within the data? Are they capable of accurately encoding topological information? Hypothetically, a good embedding should preserve both local and global structure in the graph while conveying maximum information. Beyond the graph setting, the need to benchmark new unsupervised approaches is underscored by the multiplication of recent publications in applied domains providing tips for performing a dimensionality reduction [23] or comparing existing tools [2, 20, 32]: with the rapidly increasing number of new methods, it becomes difficult to know which one to adopt — particularly when these methods can be quite sensitive. Many applications — particularly in biology — have a long track record of using graph-based techniques such as UMAP [21] and t-SNE [30] for dimensionality reduction. These methods have been well established, understood, and have been vetted by the myriad of applications and benchmarking tasks, including manifold learning and classification tasks, to which they have been deployed. On the other hand, while GNNs offer the potential for richer representation learning by capturing both feature and node information, these methods have not yet attained a similar level of reliability. This gap underscores (a) the need for a comparison of GNN-based approaches to current methods, allowing practitioners to place GNNs in the landscape of unsupervised learning methods. While GNNs offer more flexibility as they allow combining both graph data and node covariates, this comparison is nevertheless indispensable to start evaluating them as dimensionality reduction techniques. And (b) the need for more in-depth analysis and validation of GNNs in a variety of settings that go beyond node classification benchmarks, and incorporate a wider variety of tasks – including learning manifolds.

Contributions. To fill this gap, we propose here a systematic comparison of learned node representations, focusing more specifically on the context of dimensionality reduction and data visualization. To bridge classical dimensionality reduction methods that do not incorporate node features with state-of-the-art GNN-based approaches, we introduce an unsupervised learning technique inspired by UMAP, named GNUMAP, which integrates the UMAP framework for learning low-dimensional embeddings while being less parameter-intensive compared to contrastive learning method. This approach, aligning with UMAP's learning objective, facilitates a comparative analysis of different learning styles, such as self-supervised versus reconstruction-based methods. We then propose to evaluate existing methods on two tasks: (a) a suite of manifold learning tasks, thereby allowing us to compare GNN methods with more established benchmarks; and (b) classification tasks; Our goal is to provide practitioners with a clearer understanding of effective approaches in the unsupervised application of GNNs, thereby promoting their broader use across various applications.

2 GNUMAP: BRIDGING CLASSICAL DIMENSIONALITY REDUCTION WITH THE POWER OF GRAPH NEURAL NETWORKS

In this section, we introduce Graph-Neural UMAP (GNUMAP), a method that allows us to bridge the Graph Neural Network (GNN) framework with established dimensionality reduction techniques proven effective on real-world data. We begin by briefly reviewing UMAP[21], before highlighting the link with our method.

UMAP. UMAP, recognized as a standard technique for dimensionality reduction of Euclidean data denoted as $\{X_i\}_{i=1}^{n}$, employs graph-based constructs. It begins by forming a high-dimensional topological representation by assembling fuzzy simplicial sets, essentially creating a neighborhood graph. The connection probability between nodes in this graph is given by the formula $p_{ij} = p_{i|j} + p_{j|i} - p_{j|i} \times p_{i|j}$, where $p_{j|i} = \exp\left(-\frac{d(x_i,x_j)-\rho_i}{\sigma_i}\right)$. In this context, x_i denotes the original high-dimensional coordinates of

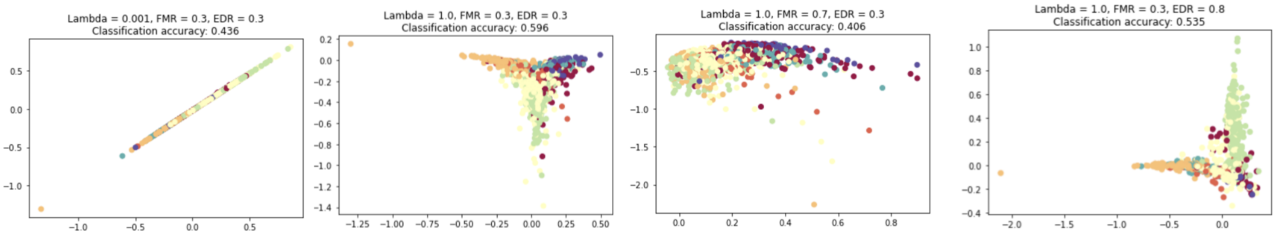

Figure 1. Node representation learning for Cora. Colours represent classes. Classification accuracy was established by running a support vector machine classifier on the learned 2D node representations, using 5-fold cross-validation to fix the kernel bandwidth. We note a substantial variation in embedding quality as the parameters vary.

input data point i, ρ_i represents the distance to the nearest neighbor of point i, and σ_i signifies the local density factor around this point.

UMAP's core process is to discover a low-dimensional representation that minimizes the cross-entropy between the high and low-dimensional topological structures. In the reduced dimensions, the connection probability analogous to p_{ij} is expressed as $q_{ij} = \frac{1}{1+\alpha\times d(y_i,y_j)^{2\beta}}$, where y_i represents the low-dimensional coordinates of data point i, with α and β being constants (specifically, $\alpha = 1.57$ and $\beta = 0.89$).

The algorithm then computes the cross-entropy loss between the pairwise connection probabilities in both high (p_{ij}) and low (q_{ij}) dimensions. This cross-entropy loss is formulated as:

$$\mathcal{L} = -\sum_i \sum_j \left[p_{ij} \log(q_{ij}) + (1 - p_{ij}) \log(1 - q_{ij}) \right]$$

This approach enables UMAP to effectively reduce dimensionality while preserving the intrinsic topological structure of the data.

GNUMAP. By contrast, in our proposed adaptation of UMAP to the graph setting, the input of the algorithm is naturally a graph where the edge weights denote the probability of node connection. This inherent graph structure of the input allows us to bypass the initial step of traditional UMAP, which converts data into a graph format. Similar to UMAP, our approach focuses on achieving dimensionality reduction by identifying a low-dimensional representation that preserves the topology of the original graph.

(a) High Dimensional Node Connectivity. Let $\mathbf{P} \in \mathcal{R}^{n\times n}$ be the input graph's (possibly weighted) adjacency matrix where p_{ij} denotes the probability of high dimensional connection between node i and j. $\mathbf{P}$ is a sparse matrix since $p_{ij} = 0$ when there is no edge between node i and j in the input graph.

(b) Low Dimensional Node Connectivity. Let y_i denote the embedding coordinate of datapoint i given as the outcome of a GNN. Then,

$$q_{ij} = \frac{1}{1 + \alpha \times d(y_i, y_j)^{2\beta}}$$

denotes the probability of the low-dimensional node connections. The derivation of q_{ij} draws significant inspiration from UMAP. Like UMAP, appropriate values of a and b are learned from the data: a and b are learned by curve fitting so that the shape of the curve produced by the distribution of distances in the high-dimensional space matches the low-dimensional one. However, while UMAP assigns a very low relative distance q_{ij} for distant nodes by setting $a = 1.57, b = 0.89$, GNUMAP is more lenient. See the comparison between GNUMAP and UMAP's calculation of q_{ij} as a function of $d(y_i, y_j)$ in Appendix 6. The a, b parameter values in GNUMAP discourage the algorithm from excessively reflecting the local connectivity, as we would prioritize global structure preservation over local in GNN embeddings. For scalability, GNUMAP computes q_{ij} only for positive edges where $p_{ij} > 0$, and an equal number of negative edges.

(c) Loss Calculation. GNUMAP employs cross-entropy loss between high and low dimensional node connectivity. However, unlike UMAP, GNUMAP weighs the attraction term by 30, so the loss becomes

$$\mathcal{L} = -\frac{1}{n} \sum_i \sum_j \left[30 p_{ij} \log(q_{ij}) + (1 - p_{ij}) \log(1 - q_{ij}) \right]$$

where n denotes the number of positive edges in the input graph. Division by n is only a stylistic choice, as it allows more intuitive monitoring of GNUMAP loss over epochs. However, a significant difference between UMAP cross entropy and that of GNUMAP is the attraction term weight of 30. This alteration encourages GNUMAP to more severely penalize the lack of connection in the embeddings despite an existing edge in the original graph. We provide in Appendix 7 a visualization of the UMAP and GNUMAP loss function. This design choice was made to prioritize the preservation of high-dimensional connection. The procedure is summarized in Algorithm 1 below.

Our motivation for introducing this method here is purely pragmatic and concerned with its broader applicability in across various domains. Consequently, the proposed approach is simple, requiring solely the specification of the minimum distance between nodes in embedding space — a parameter that we hope practitioners to be comfortable with.

Remark 1. We note the similarity between the proposed embedding technique and traditional auto-encoders and variational auto-encoder methods. The latter, like UMAP, are reconstruction-based techniques — they aim to find a low-dimensional representation of the data such that the distance between the learned node embeddings is predictive of the existence of an edge between nodes in the original graph. One of the main differences between our proposed architecture and these traditional auto-encoder approaches is that the appropriate distance between low-dimensional embeddings Q is inferred from the data, but does not have to be learned. This allows us to compare reconstruction-based methods with more traditional methods on a more equal basis. Moreover, we found that our rebalancing of the loss corresponding to negative edges and

Algorithm 1 GNUMAP Algorithm for Output Dimension = D

```
Inputs: Adjacency matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$;
        $n_{\text{pos}} = \text{count}(p_{ij} > 0 \text{ in } \mathbf{P})$;
        $E_{pos} = \{(i,j) \in \mathcal{V} \times \mathcal{V} : \text{ such that } p_{ij} > 0\}$.
        GNN architecture for embedding nodes in dimension $D$;
for number of epochs = 400 do
    Step 1- Sample $n_{\text{neg}}$ negative edges:
        $E_{neg} = \{(i,j) \in \mathcal{V} \times \mathcal{V} : \text{ such that } p_{ij} = 0\}$.
    Step 2- Compute node embeddings:
        $\mathcal{Y}^{n \times F}$ = GNN(Features, Edge Index);
    Step 3- Compute $d(y_i, y_j)$ for all sampled negative edges in $E_{neg}$;
    Step 4- Compute $d(y_i, y_j)$ for all positive edges $(i,j) \in E_{pos}$
    Step 5- Append results from steps 3 and 4 to compute $D$ and corresponding loss:
    $Q = \frac{1}{1+\alpha D^{2\beta}}$ to get $Q \in \mathbb{R}^{1 \times 2 * n_{\text{positive}}}$
    Step 5- Access $\mathbf{P}$ at positive edge indices and negative sampled indices to get $P \in \mathbb{R}^{1 \times 2 * n_{\text{positive}}}$
    Step 6- Compute $\mathcal{L}$ = Weighted Cross-Entropy($P, Q$) & Backpropagate $\mathcal{L}$
end for
```

positive edges led to better representation — allowing us to avoid phenomena similar to node collapse in VAEs.

Remark 2. Contrary to self-supervised techniques for node embeddings, this proposed method does not rely on the choice of specific hyperparameters to perform well.

Remark 3. The reconstruction objectives are typically more amenable to homophilic networks (where adjacent nodes are presumed to be similar) rather than heterophilic networks — a limitation of our framework compared to self-supervised learning techniques.

Remark 4. The closest method related to this direct extension of UMAP is SpaGCN [14]. Developed for a transcriptomics application, SpaGCN is a GCN algorithm that incorporates histological data to identify spatial domains and the associated gene expressions. From methods perspective, while SpaGCN is also an adaptation of UMAP to the graph setting, it differs from our approach in two significant ways: (i) SpaGCN directly targets creating a prespecified number of clusters in the embedding space. The associated clusters are selected by minimizing the KL divergence between their distance in low-dimensional space, and their distance in high-dimensional space; The definitions of q and p are also quite different: SpaGCN's q is assigned to a lower value when the embeddings are further from the cluster centers, and p is updated as the twice-normalized square of q at every three epochs. GNUMAP, by contrast, is amenable to learning a wider variety of manifolds, as it does not explicitly target the clustering of embeddings, but simply seeks to reconstruct a low-dimensional representation of the nodes that aligns with the original graph.

3 EVALUATING GNNS' MANIFOLD LEARNING ABILITIES

Having established a natural extension of UMAP to the GNN setting, we propose to benchmark these approaches on a set of toy examples evaluating their ability to correctly learn an underlying manifold structure. While this is a standard test for any dimensionality reduction technique, GNNs have not yet been evaluated in this specific context.

3.1 Metrics

Current approaches to evaluating embeddings generally fall into one of two types: (a) visual methods, where individuals visually inspect the learned embeddings to assess if they align with their expectations, and (b) classification-based methods. In the classification-based approach, it is presumed that there exists a set of labels, not used during training, which effectively represent the structure of the underlying data. In our work, we extend these latter metrics for application to manifold scenarios (which involve continuous structures rather than discrete ones) and quantify the performance of the learned representations using the following metrics.

General Metrics To assess the embedding quality, we propose measuring the agreement between the learned embedding space and the original data. This agreement can in particular be measured using:

- **Agreement in Local Geometry** We compute the percentage of overlap between the original graph's k-nearest neighbors and knn graphs on the embedding space. A higher percentage denotes better local geometry preservation.
- **Spearman Correlation** Inspired by a methodology introduced in [1], we compute the distance matrix of the input graph, then evaluate the Spearman correlation between the low and high dimensional pairwise distance matrix. This metric evaluates the relationship between learned embedding space and the original graph using a monotonic function. The metric spans [-1,1] where -1 denotes a relationship modeled by a perfectly decreasing monotonic function, and vice versa. A higher Spearman correlation implies a similarity between the learned embedding and the original data.
- **Classification Accuracy** While our embeddings are unsupervised, we assume quality embeddings would form well-separated and helpful features for SVM multi-class classification problem. Consequently, we divide our manifold into different clusters (in intrinsic space), and assess the accuracy of the embedding space in recovering the different clusters. Higher accuracy thus implies informative embeddings.
- **Frechet Distance** Finally, we implemented the Frechet inception distance [12], a metric originally proposed to measure the similarity between images, to evaluate the 2d Frechet distance between the learned embedding coordinates and the original data. Smaller Frechet distance denotes better similarity between embedding and the original data coordinates.

Metrics for Clustered Data For data that are expected to form clusters (e.g. synthetic blobs, citation networks, Mouse Spleen gene expression prediction), we propose metrics evaluating the quality of cluster separation in the embedding space.

- **Davies Bouldin Score** This metric is the average ratio of inter-cluster distance to intra-cluster distance to the most similar cluster[9]. Davies Bouldin score is lower when clusters are concentrated and far apart. A lower score denotes better cluster separation, and the minimum score is zero.
- **Calinski Harabasz Score** This metric is the ratio of the sum of intra-cluster dispersion and of inter-cluster dispersion[5]. Calinski Harabasz score is higher when the intra-cluster variability is low and inter-cluster variability is high. A higher score indicates better score cluster separation.
- **Silhouette Score** This metric is the mean silhouette coefficient[25] of all data points. Let a = mean intra-cluster distance and b = mean nearest-cluster distance. Then, the silhouette coefficient for each datapoint is computed by $\frac{b-a}{\max(a,b)}$. Silhouette score is concerned with how similar a data point is to its assigned cluster compared to other clusters. The best value, 1 indicates well-separated and correctly assigned clusters, while the worst value, -1, often indicates that a sample is more similar to a different cluster. A silhouette score of 0 indicates overlapping clusters.

3.2 Synthetic Geometric Datasets

With the proposed metrics, we evaluate GNUMAP performance on four synthetic geometric graph datasets: Blobs, Swissroll, Moons, and Circles. The datasets were generated with ground-truth low-dimensional coordinates and cluster labels, which allows us to calculate our proposed metrics and also to inspect if our metrics are coherent with the embedding visualization. The four datasets were generated with 1000 nodes, each connected to its 50 nearest neighbors. The features were instantiated as the embeddings of a 10-component Laplacian eigenmap decomposition of the induced 50-nearest graph. The latter is indeed an established procedure to instantiate features on a graph[14]. We compare GNUMAP with state-of-the-art GNN embedding methods tailored for dimensionality reduction: DGI[31], BGRL[29], CCA-SSG[34], and SpaGCN [14]. We also compare with well-established dimensionality reduction methods PCA, t-SNE[30], Isomap[28], Laplacian Eigenmap[4], UMAP[21], and DenseMAP[22]. Note all methods mentioned above are Euclidean methods, which enable a consistent accessment with respect to our proposed metrics.

We implemented a standard 2-layer GCN architecture for DGI, BGRL, and CCA-SSG. BGRL requires hyperparameters feature mask rate p_m and edge drop rate p_e, and CCA-SSG requires p_m, p_e, λ. On synthetic datasets, we visually inspected embedding quality over multiple hyperparameter combinations, and we chose to incorporate $\lambda = 1e-5$, edge drop rate = 0.5, feature drop rate = 0 into the experimental setup as they showed one of the best embedding visualizations. Source code for CCA-SSG includes λ, edge drop rate, and feature drop rate for Cora and Pubmed, so we followed such hyperparameter decisions with Cora and Pubmed experiments. However, for mouse spleen data, which is without previous work with CCA-SSG, we incorporated the same hyperparameters we used for synthetic datasets. Finally, SPAGCN implementation is a 1-layer GCN with feature mask rate $p_m = 0$ to follow the original model hyperparameters.

We visualize the resulting embeddings in Figure 2. For PCA, t-SNE, Isomap, Laplacian Eigenmap, and DenseMAP visualizations, refer to Appendix 8. Furthermore, we compute proposed metrics over 100 experiment and summarize the results in Figure 3.

Discussion. Overall, GNUMAP compares favorably to the most used dimensionality reduction techniques in these small examples. It consistently outperforms other self-supervised GNN-based methods.

Synthetic Swissroll GNUMAP outperforms all state-of-the-art GNN-based methods according to our proposed metric in 3, and the GNUMAP embedding visualization in 2 is coherent with the metrics, as GNUMAP successfully unrolls the swissroll to closely resemble the ground truth manifold. Also, in comparison to traditional dimensionality reduction methods, GNUMAP outperforms them for Calinski-Harabasz score, Spearman correlation with the original graph, and the overlap percentage of 50 nearest neighbors. Further on the comparison with traditional dimensionality reduction methods, GNUMAP has similar or smaller frechet distance. Isomap is the only dimensionality reduction method that performs better than GNUMAP on classification accuracy, Silhouette score, and Davies Bouldin score.

Synthetic Blobs GNUMAP outperforms other GNN methods in all metrics except for Calinski Harabasz score, which is outperformed by CCA-SSG, but the standard deviation for CCA-SSG is very large. Such findings are coherent with the embedding visualizations, as GNUMAP clearly distinguishes the four clusters. CCA-SSG embeddings also had clearly established blobs, but embeddings were concentrated in almost a line, implying a potential oversmoothing problem. On the other hand, when compared with traditional dimensionality reduction methods, GNUMAP is on par or outperformed on all metrics. Blobs appear to be a dataset better captured by traditional dimensionality reduction, and the expressivity of graph neural networks does not seem to improve embedding quality in this case.

4 EVALUATING UNSUPERVISED GNNS' EMBEDDING PERFORMANCE ON REAL-WORLD DATA

To assess GNUMAP's performance across real-world data, we conducted a comparative analysis using well-established graph benchmark datasets: Cora and Pubmed. They are standard citation network benchmark datasets.[33] In these networks, nodes represent scientific publications, and edges denote citation links between publications. Node features are the bag-of-words representation of papers, and node label is the academic topic of a paper.

In addition, we incorporated Mouse Spleen cell type annotation data[11] to access GNUMAP performance in biological applications. The dataset was originally generated by "co-detection by indexing"(CODEX) techniques, which is to take the image of a tissue section, and at each of the tissue locations, the relevant biomarkers information is measured and recorded[11]. While the 3-topic cluster labels in the Mouse Spleen dataset are generated by the spatial LDA model, previous work[7] has established that such topic annotations are biologically consistent with immunologist-labeled topics. Therefore, we will regard the 3-topic cluster labels as ground truth labels in the Mouse Spleen dataset for the subsequent analysis.

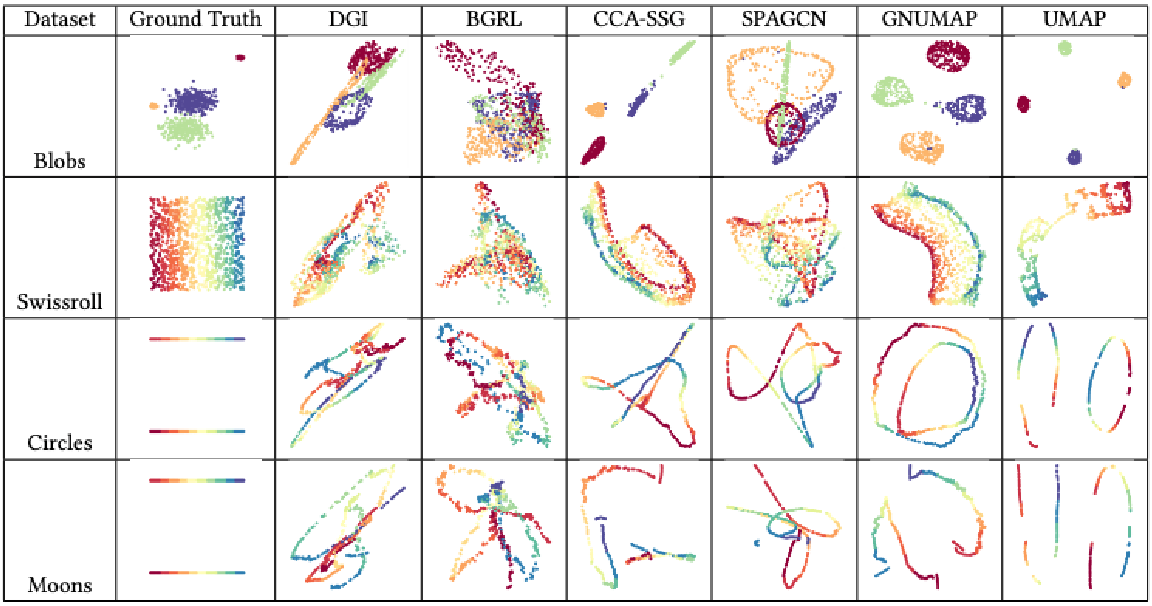

Figure 2. Node representation learning for synthetic datasets Blobs, Swissroll, Circles, Moons. Colours represent assigned ground truth cluster label.

For all experiments with real-world datasets, GNUMAP was compared against CCA-SSG, SPAGCN, and UMAP for the following purposes:

(1) Comparison with state-of-the-art GNN method for dimensionality reduction.
(2) Comparison with state-of-the-art GNN methods for biomedical applications.
(3) Comparison with traditional dimensionality reduction methods.

All datasets have class labels, enabling a comparison between embedding visualization and our calculated metrics. For scalable metric evaluation, we randomly sampled 2000 datapoints and calculated the following metrics: classification accuracy, Silhouette score, Calinski Harabasz score, and Davies Bouldin Score. These metrics convey the general degree of embedding informativity and the quality of clustering, which is appropriate for evaluating Cora, Pubmed, and Mouse Spleen data as one expects some clustering among the same classes. We also measured the time for the full training cycle, enabling a comparison of GNUMAP's runtime with that of CCA-SSG, SPAGCN, and UMAP, thus demonstrating the empirical scalability of the GNUMAP algorithm. The results are presented in Figure 4 and Table 5.

Discussion.

Cora While GNUMAP outperforms all other models in classification accuracy, its performance across other metrics are worse than that of UMAP. UMAP performs consistently well across all 4 metrics, while SPAGCN was the worst model in terms of the metrics. Such was reflective in the embedding visualizations, where all models except for SPAGCN exhibited defined clusters. Interestingly, CCA-SSG displayed a very high Calinski Harabasz score, but such trend did not continue for other clustering-related metrics Silhouette score and Davies Bouldin. This suggests our proposed metrics should be used in conjunction with another, as dependence on a single metric may lead to misleading conclusions on the embedding quality. With regards to the scalability of GNUMAP on the Cora dataset with 2708 nodes and 10556 undirected edges, GNUMAP was faster than CCA-SSG but slower than UMAP and SPAGCN. SPAGCN's fast runtime is due to its early stopping condition. For future work, GNUMAP may implement early stopping conditions without compromising the quality of embeddings. We emphasize as well that this reconstruction-based method seems scalable. (Its application to the MOUSE dataset with 81k nodes and 467k edges only took 2139 seconds on average, an improvement of 25.2% over competing method CCA-SSG.)

Pubmed SPAGCN was notably performing the worst across all four metrics, but CCA-SSG, GNUMAP, and UMAP performed similarly with respect to the metrics despite the distinct style of embeddings. This highlights the limits of sole visual inspection of embedding quality. For example, CCA-SSG exhibits more compact embeddings that could hint for low embedding quality, but further inspection on cluster separation and informativity of embeddings through our proposed metrics suggest otherwise.

Mouse Spleen While CCA-SSG, GNUMAP, and SPAGCN perform similarly across all four metrics, UMAP performance lags behind on all metrics except for silhouette score by a small margin. This result shows the limitation of traditional dimensionality reduction methods in domains where expressive aggregation of neighborhood information is beneficial, such as molecular dynamics or cell transcriptomics tasks. This finding is coherent with the embedding visualizations, where CCA-SSG, GNUMAP, and SPAGCN successfully separate biologically meaningful cluster labels, but UMAP is returning an embedding similar to a slice or the original input data.

	Model	Classification Accuracy	Silhouette Score	Calinski Harabasz Score	Davies Bouldin Score	Frechet Distance	Spearman Correlation w/ Original Graph	Overlap % of 50 Neighbours	Time(sec)
Swissroll	DGI	0.22 ± 0.03	-0.24 ± 0.03	126.98 ± 42.1	6.93 ± 2.14	0.04 ± 0.01	0.24 ± 0.11	0.37 ± 0.04	184 ± 9
	BGRL	0.25 ± 0.02	-0.21 ± 0.03	137.9 ± 31.67	5.8 ± 1.13	0.03 ± 0.01	0.27 ± 0.11	0.4 ± 0.03	299 ± 31
	CCA-SSG	0.22 ± 0.05	-0.14 ± 0.05	572.63 ± 164.94	5.23 ± 1.72	0.06 ± 0.03	0.69 ± 0.15	0.52 ± 0.05	79 ± 13
	SPAGCN	0.26 ± 0.02	-0.17 ± 0.03	180.36 ± 34.05	5.2 ± 1.5	0.02 ± 0.02	0.29 ± 0.08	0.46 ± 0.03	4 ± 0
	GNUMAP	0.34 ± 0.04	0.01 ± 0.04	691.95 ± 147.13	2.95 ± 0.58	0.01 ± 0.01	0.93 ± 0.03	0.7 ± 0.04	109 ± 12
	PCA	0.19 ± 0.05	-0.22 ± 0.04	127.02 ± 9.55	11.18 ± 5.32	0.0 ± 0.0	0.07 ± 0.04	0.2 ± 0.03	0 ± 0
	Laplacian Eigenmap	0.14 ± 0.02	-0.29 ± 0.02	158.42 ± 12.17	14.48 ± 2.55	0.02 ± 0.01	0.0 ± 0.01	0.18 ± 0.01	0 ± 0
	Isomap	0.42 ± 0.03	0.12 ± 0.03	584.3 ± 86.23	1.61 ± 0.63	0.01 ± 0.0	0.41 ± 0.01	0.58 ± 0.03	0 ± 0
	TSNE	0.31 ± 0.02	-0.01 ± 0.03	330.03 ± 109.85	3.97 ± 1.65	0.01 ± 0.0	0.31 ± 0.06	0.39 ± 0.01	6 ± 0
	UMAP	0.25 ± 0.03	-0.07 ± 0.03	551.88 ± 228.41	5.25 ± 1.89	0.03 ± 0.02	0.34 ± 0.04	0.37 ± 0.02	9 ± 0
	DenseMAP	0.23 ± 0.02	-0.1 ± 0.02	305.49 ± 98.86	5.51 ± 1.56	0.04 ± 0.02	0.32 ± 0.04	0.37 ± 0.02	9 ± 1
Blobs	DGI	0.73 ± 0.08	0.06 ± 0.08	324.12 ± 286.32	5.67 ± 16.57	0.1 ± 0.06	0.3 ± 0.12	0.35 ± 0.05	225 ± 18
	BGRL	0.75 ± 0.08	0.08 ± 0.08	238.26 ± 220.92	4.02 ± 3.29	0.07 ± 0.04	0.28 ± 0.12	0.38 ± 0.04	362 ± 42
	CCA-SSG	0.96 ± 0.05	0.59 ± 0.15	10339.78 ± 7848.12	0.73 ± 0.51	0.13 ± 0.09	0.56 ± 0.17	0.48 ± 0.05	97 ± 19
	SPAGCN	0.78 ± 0.06	0.05 ± 0.07	140.21 ± 83.26	4.95 ± 4.33	0.06 ± 0.04	0.24 ± 0.08	0.44 ± 0.04	4 ± 0
	GNUMAP	0.99 ± 0.03	0.65 ± 0.1	3296.67 ± 1013.86	0.55 ± 0.29	0.05 ± 0.03	0.66 ± 0.13	0.62 ± 0.02	131 ± 18
	PCA	0.99 ± 0.02	0.77 ± 0.09	9433.98 ± 6156.32	0.39 ± 0.21	0.06 ± 0.04	0.81 ± 0.07	0.67 ± 0.06	0 ± 0
	Laplacian Eigenmap	0.9 ± 0.12	0.79 ± 0.23	8.75e+30 ± 9.25e+30	0.03 ± 0.09	0.12 ± 0.07	0.71 ± 0.17	0.44 ± 0.05	0 ± 0
	Isomap	0.99 ± 0.02	0.79 ± 0.09	11693.2 ± 8607.31	0.38 ± 0.26	0.07 ± 0.05	0.8 ± 0.07	0.58 ± 0.06	1 ± 0
	TSNE	0.99 ± 0.02	0.72 ± 0.05	4948.24 ± 855.28	0.4 ± 0.1	0.05 ± 0.03	0.76 ± 0.06	0.75 ± 0.01	6 ± 0
	UMAP	0.99 ± 0.02	0.86 ± 0.08	29225.18 ± 10329.49	0.22 ± 0.15	0.06 ± 0.04	0.8 ± 0.08	0.73 ± 0.03	9 ± 1
	DenseMAP	0.99 ± 0.02	0.81 ± 0.07	10594.87 ± 3265.26	0.32 ± 0.16	0.05 ± 0.03	0.79 ± 0.06	0.67 ± 0.02	9 ± 1

Figure 3. Evaluated mean and standard deviation of proposed metrics for synthetic datasets Swissroll and Blobs over 100 experiments. The upward arrows next to metric name denote a better performance with higher metrics, whereas the downward arrows denote better performance with lower metrics.

5 CONCLUSION

With the exponential growth in data complexity, statisticians and computer scientists have recognized graphs as a modeling framework for high-dimensional data types. Therefore, modeling with graphs, predominantly through GNNs, is gaining traction in various areas, especially the biological[7, 14] domain. Through this work, we contribute to the ongoing research on unsupervised graph learning by proposing GNUMAP: a GNN inspired by theoretical research on the dimensionality reduction technique UMAP. Through

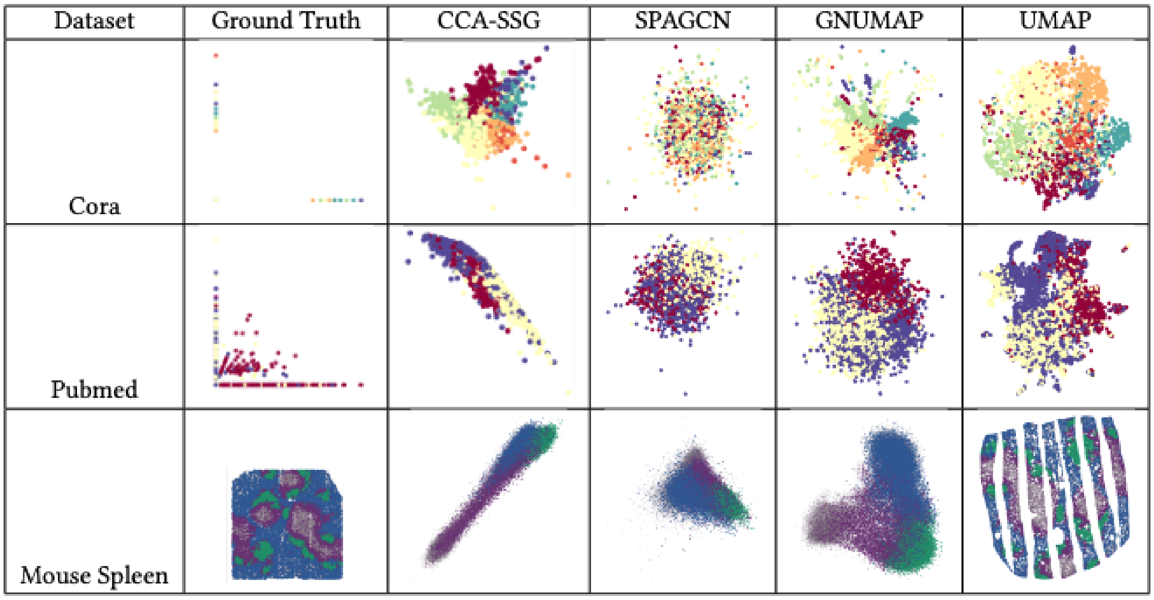

Figure 4. Node representation learning for real-world datasets Cora, Pubmed, Mouse Spleen. Colours represent assigned ground truth cluster label. In Mouse Spleen data, blue denotes B-cells, purple denotes marginal zone B-cells, gray denotes non-B cells, and green denotes red pulp[7]

	Model	Classification Accuracy ↑	Silhouette Score ↑	Calinski Harabasz Score ↑	Davies Bouldin Score ↓	Time (seconds)
Cora	CCA-SSG	0.57 ± 0.06	-0.06 ± 0.05	423.88 ± 102.59	6.21 ± 7.48	407 ± 50
	GNUMAP	0.69 ± 0.03	-0.01 ± 0.02	155.51 ± 31.87	4.55 ± 2.62	66 ± 1
	SPAGCN	0.32 ± 0.01	-0.12 ± 0.02	19.52 ± 8.46	16.62 ± 7.97	4 ± 0
	UMAP	0.68 ± 0.0	0.06 ± 0.0	258.1 ± 0.0	2.89 ± 0.0	23 ± 1
Pubmed	CCA-SSG	0.63 ± 0.04	0.02 ± 0.03	258.31 ± 110.91	3.69 ± 2.69	506 ± 26
	GNUMAP	0.69 ± 0.02	0.08 ± 0.02	313.17 ± 58.91	3.01 ± 0.85	392 ± 43
	SPAGCN	0.46 ± 0.03	-0.02 ± 0.01	35.98 ± 25.23	10.7 ± 6.76	48 ± 9
	UMAP	0.67 ± 0.0	0.06 ± 0.0	227.68 ± 0.0	2.71 ± 0.0	54 ± 3
Mouse Spleen	CCA-SSG	0.67 ± 0.03	-0.05 ± 0.05	281.46 ± 61.33	2.3 ± 0.97	2678 ± 1018
	GNUMAP	0.66 ± 0.0	-0.04 ± 0.01	294.62 ± 24.74	1.96 ± 0.05	2139 ± 158
	SPAGCN	0.63 ± 0.03	-0.04 ± 0.04	213.17 ± 87.27	2.79 ± 0.61	421 ± 51
	UMAP	0.55 ± 0.0	-0.02 ± 0.0	2.13 ± 0.0	38.41 ± 0.0	216 ± 75

Figure 5. Summary mean and standard deviation of proposed metrics over multiple experiments with real-world datasets Cora, Pubmed, Mouse Spleen.

our unique metrics that evaluate informativeness (classification accuracy), preservation of original graph information (agreement in local geometry, spearman correlation, frechet distance), and cluster quality (Davies-Bouldin score, Calinski-Harabasz score, silhouette score), we establish GNUMAP as a robust and expressive GNN embedding method that performs outperforms many existing GNN embedding methods, as well as dimensionality reduction methods depending on the task. Clear clustering of GNUMAP embeddings across synthetic geometric datasets as well as Cora, Pubmed, and Mouse Spleen further highlights the adaptive power of GNUMAP. GNUMAP performance can be characterized as "data-agnostic"– unlike state-of-the-art GNNs, there is minimal hyperparameter tuning required to enhance GNUMAP performance, allowing an easy application to any dataset.

Limitations and Future Work The reconstruction-based approach that we propose here in this paper, GNUMAP, is by design extremely simple, and its objective is interpretable. While it performs well on the examples that we provide here, this method is only valid for homophilic datasets — where edges encode similarities between nodes.

However, this method also lends itself to natural extensions — such as accounting for cluster densities using the density coefficient of DenseMAP[22]. This would allow a better capture of the characteristics of the data in the low-dimensional embedding.

REFERENCES

[1] Asif Adil, Vijay Kumar, Arif Tasleem Jan, and Mohammed Asger. 2021. Single-cell transcriptomics: current methods and challenges in data acquisition and analysis. *Frontiers in Neuroscience* 15 (2021), 591122.

[2] Ashley Babjac, Taylor Royalty, Andrew D Steen, and Scott J Emrich. 2022. A comparison of dimensionality reduction methods for large biological data. In *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 1–7.

[3] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. 2018. Relational inductive biases, deep learning, and graph networks. https://doi.org/10.48550/ARXIV.1806.01261

[4] Mikhail Belkin and Partha Niyogi. 2003. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation* 15, 6 (2003), 1373–1396. https://doi.org/10.1162/089976603321780317

[5] T. Caliński and J Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics* 3, 1 (1974), 1–27. https://doi.org/10.1080/03610927408827101 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/03610927408827101

[6] Sergio Casas, Cole Gulino, Renjie Liao, and Raquel Urtasun. 2019. Spatially-aware graph neural networks for relational behavior forecasting from sensor data. *arXiv preprint arXiv:1910.08233* (2019).

[7] Zhenghao Chen, Ilya Soifer, Hugo Hilton, Leeat Keren, and Vladimir Jojic. 2020. Modeling Multiplexed Images with Spatial-LDA Reveals Novel Tissue Microenvironments. *Journal of Computational Biology* (2020).

[8] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, Xiao Dong, and Yinhai Wang. 2019. *High-order graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting*. Technical Report.

[9] David L. Davies and Donald W. Bouldin. 1979. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1, 2 (1979), 224–227. https://doi.org/10.1109/TPAMI.1979.4766909

[10] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint arXiv:1509.09292* (2015).

[11] Yury Goltsev, Nikolay Samusik, Julia Kennedy-Darling, Salil Bhate, Matthew Hale, Gustavo Vazquez, Sarah Black, and Garry P Nolan. 2018. Deep profiling of mouse splenic architecture with CODEX multiplexed imaging. *Cell* 174, 4 (2018), 968–981.

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium. *CoRR* abs/1706.08500 (2017). arXiv:1706.08500 http://arxiv.org/abs/1706.08500

[13] Ilgee Hong, Huy Tran, and Claire Donnat. 2023. A Simplified Framework for Contrastive Learning for Node Representations. *arXiv preprint arXiv:2305.00623* (2023).

[14] Jian Hu, Xiangjie Li, Kyle Coleman, Amelia Schroeder, Nan Ma, David J Irwin, Edward B Lee, Russell T Shinohara, and Mingyao Li. 2021. SpaGCN: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature methods* 18, 11 (2021), 1342–1351.

[15] Satoki Ishiai, Ikki Yasuda, Katsuhiro Endo, and Kenji Yasuoka. 2024. Graph-Neural-Network-Based Unsupervised Learning of the Temporal Similarity of Structural Features Observed in Molecular Dynamics Simulations. *Journal of Chemical Theory and Computation* (2024).

[16] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. https://doi.org/10.48550/ARXIV.1609.02907

[17] Junyi Li, Wei Jiang, Henry Han, Jing Liu, Bo Liu, and Yadong Wang. 2021. ScGSLC: an unsupervised graph similarity learning framework for single-cell RNA-seq data clustering. *Computational Biology and Chemistry* 90 (2021), 107415.

[18] Michelle M Li, Kexin Huang, and Marinka Zitnik. 2021. Representation Learning for Networks in Biology and Medicine: Advancements, Challenges, and Opportunities. *arXiv preprint arXiv:2104.04883* (2021).

[19] Tengfei Ma, Junyuan Shang, Cao Xiao, and Jimeng Sun. 2019. GENN: predicting correlated drug-drug interactions with graph energy neural networks. *arXiv preprint arXiv:1910.02107* (2019).

[20] Tamasha Malepathirana, Damith Senanayake, Rajith Vidanaarachchi, Vini Gautam, and Saman Halgamuge. 2022. Dimensionality reduction for visualizing high-dimensional biological data. *Biosystems* 220 (2022), 104749.

[21] Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 [stat.ML]

[22] Ashwin Narayan, Bonnie Berger, and Hyunghoon Cho. 2020. Density-Preserving Data Visualization Unveils Dynamic Patterns of Single-Cell Transcriptomic Variability. *bioRxiv* (2020). https://doi.org/10.1101/2020.05.12.077776 arXiv:https://www.biorxiv.org/content/early/2020/05/14/2020.05.12.077776.full.pdf

[23] Lan Huong Nguyen and Susan Holmes. 2019. Ten quick tips for effective dimensionality reduction. *PLoS computational biology* 15, 6 (2019), e1006907.

[24] Gabriele Partel and Carolina Wählby. 2021. Spage2vec: Unsupervised representation of localized spatial gene expression signatures. *The FEBS Journal* 288, 6 (2021), 1859–1870.

[25] Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

[26] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80. https://doi.org/10.1109/TNN.2008.2005605

[27] Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al. 2020. A deep learning approach to antibiotic discovery. *Cell* 180, 4 (2020), 688–702.

[28] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* 290, 5500 (2000), 2319–2323. https://doi.org/10.1126/science.290.5500.2319 arXiv:https://www.science.org/doi/pdf/10.1126/science.290.5500.2319

[29] Shantanu Thakoor et al. 2021. Large-Scale Representation Learning on Graphs via Bootstrapping. *arXiv preprint arXiv:2102.06514* (2021).

[30] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[31] Petar Veličković et al. 2018. Deep Graph Infomax. *arXiv preprint arXiv:1809.10341* (2018).

[32] Ruizhi Xiang, Wencan Wang, Lei Yang, Shiyuan Wang, Chaohan Xu, and Xiaowen Chen. 2021. A comparison for dimensionality reduction methods of single-cell RNA-seq data. *Frontiers in genetics* 12 (2021), 646936.

[33] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. arXiv:1603.08861 [cs.LG]

[34] Hengrui Zhang et al. 2021. From Canonical Correlation Analysis to Self-Supervised Graph Neural Networks. In *Advances in Neural Information Processing Systems*, Vol. 34. 76–89.

[35] Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S. Yu. 2021. From Canonical Correlation Analysis to Self-supervised Graph Neural Networks. https://doi.org/10.48550/ARXIV.2106.12484

[36] Ruochi Zhang, Jianzhu Ma, and Jian Ma. 2020. DANGO: Predicting higher-order genetic interactions. *bioRxiv* (2020), 2020–11.

[37] Yanqiao Zhu et al. 2020. Deep Graph Contrastive Representation Learning. *arXiv preprint arXiv:2006.04131* (2020).

[38] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34, 13 (2018), i457–i466.

A GNUMAP DESIGN CHOICES

A.1 Effect of Changing a, b

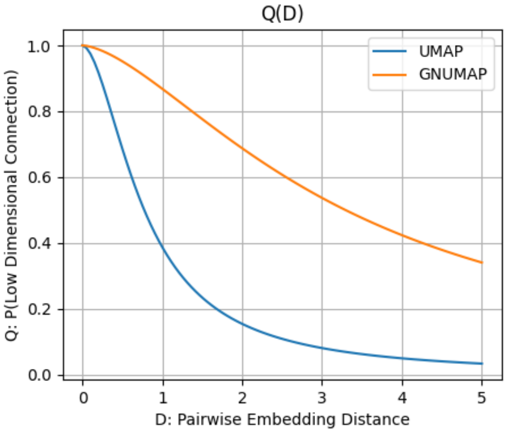

Figure 6: Probability of low dimensional connection as a function of pairwise low-dimensional distance

A.2 UMAP Cross-Entropy vs. GNUMAP Cross-Entropy

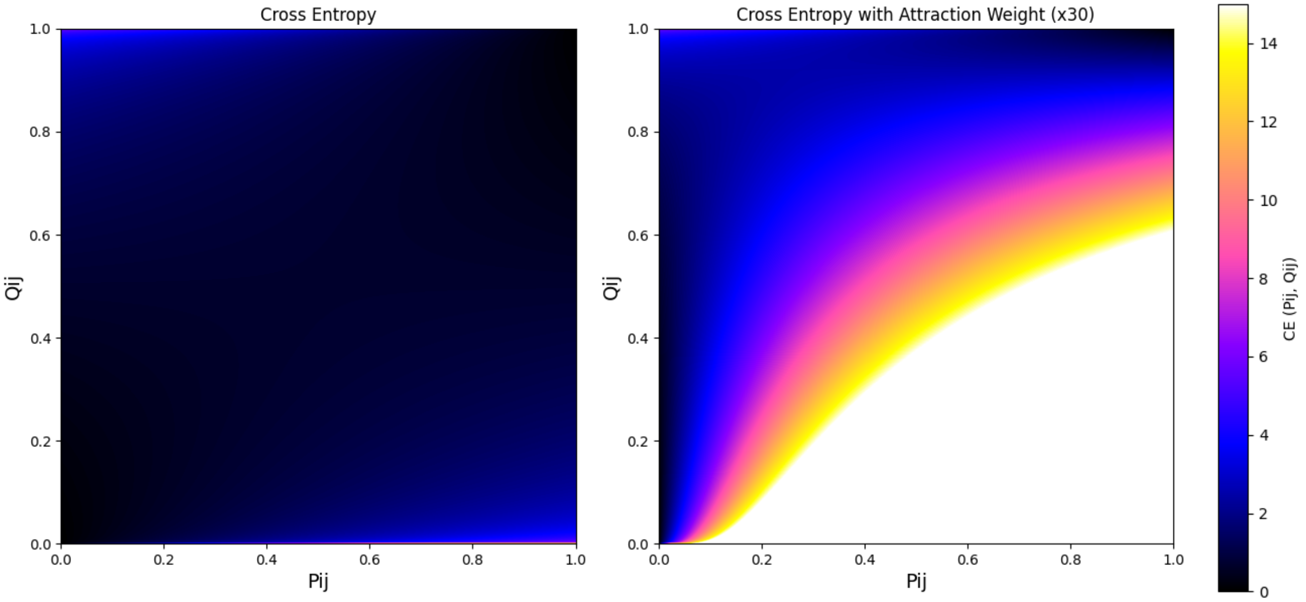

Figure 7. Cross Entropy as a function of p_{ij}, q_{ij}, where p_{ij} denotes probability of connection between datapoint i and j in the input graph, and q_{ij} denotes algorithm-assigned probability of connection between datapoint i and j in the low-dimensional space

B ADDITIONAL SYNTHETIC EXPERIMENT EMBEDDING VISUALIZATIONS

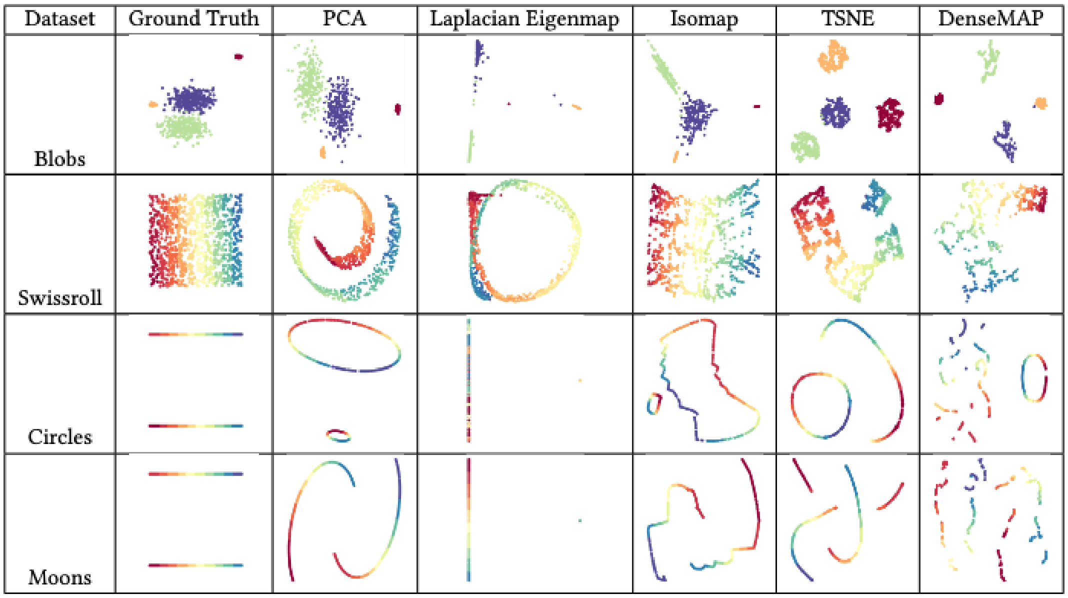

Figure 8. Node representation learning for synthetic datasets Blobs, Swissroll, Circles, Moons. Colours represent assigned ground truth cluster label.