A APPENDIX

A.1 DATASETS

We perform the experiments on ten networks of various sizes and structures. Each network is considered as undirected and unweighted for the consistency of the experiments. (i) *Cora* (Sen et al., 2008) is a citation network regarding machine learning publications which consist of seven categories. (ii) *Dblp* (Perozzi et al., 2017) is Perozzi et al. (2017) is a co-authorship network in which the node labels denote the research fields. (iii) *AstroPh* (Leskovec et al., 2007) is a collaboration network constructed by papers submitted to astrophysics category of the e-print archive, ArXiv. The nodes represent authors, and there is an edge between a pair of nodes if the corresponding authors have co-authored a paper. (iv) *GrQc* (Leskovec et al., 2007) is another co-authorship network similarly devised with the papers whose section are labeled as general relativity and quantum cosmology category. (v) *Facebook* (Leskovec & Mcauley, 2012) is a social network built with the data obtained by a survey conducted through a Facebook application. (vi) *HepTh* (Leskovec et al., 2007) is a collaboration network constructed with the works of the high energy physics category of ArXiv. (vii) *Amazon* (Yang & Leskovec, 2015a) is a network constructed by crawling the Amazon website in which the node labels show the product categories. We consider the top $5,000$ categories in the classification experiments. (viii) *YouTube* (Yang & Leskovec, 2015b; Mislove et al., 2007) is a social network built by the video-sharing platform allowing its users to form connections and groups. (ix) *Flickr* (Mislove et al., 2007) network has been crawled from the image sharing application Flickr in which links indicate the connections between the users. (x) *Flixster* (Zafarani & Liu, 2009) has been extracted from the social-networking movie platform allowing its users to share movie ratings and meet with other users sharing similar interests. The detailed statistics of the networks are provided in Table 1.

Table 1: Statistics of networks. N: number of nodes, M: number of edges, d^*: average node degree

Network	N	M	d^*
Cora	$2,708$	$5,278$	3.898
Dblp	$27,199$	$66,832$	4.914
AstroPh	$17,903$	$197,031$	22.010
GrQc	$5,242$	$14,496$	5.531
Facebook	$4,039$	$88,234$	43.691
HepTh	$8,638$	$24,827$	5.748
Amazon	$334,868$	$925,876$	5.530
YouTube	$1,138,499$	$2,990,443$	5.253
Flickr	$1,715,255$	$15,555,042$	18.137
Flixster	$2,523,386$	$7,918,801$	6.276

A.2 BASELINE METHODS

In our experiments, we have run various graph representation learning methods in order to evaluate the performance of our proposed approach. (i) DEEPWALK (Perozzi et al., 2014) that generates a set of fixed-length node sequences by performing a uniform random walking strategy and extracts the embeddings by maximizing the co-occurrence probabilities of nodes within a certain window size. (ii) NODE2VEC (Grover & Leskovec, 2016) employing the same methodology to learn the node representations, but introduces two additional parameters to manipulate the random walk more flexibly. (iii) LINE (Tang et al., 2015) that learns node embedding vectors by optimizing the first- and second-order proximity information. (iv) NETMF (Qiu et al., 2018) learns embeddings by factorizing the pointwise mutual information matrix constructed by using the node co-occurrences in the random walks. (v) NETSMF (Qiu et al., 2019) that is a scalable extension of the NETMF method (Qiu et al., 2018) which aims to learn representations by factorizing the designed pointwise mutual information matrix. (vi) RANDNE (Zhang et al., 2018) that targets large networks and extracts the embedding vectors by applying iterative Gaussian random projections. (vii) PRONE (Zhang et al., 2019) that initially learns the node representations by an efficient sparse matrix factorization and the node embeddings are enhanced with spectral propagation operations. (viii) LOUVAINNE

(Bhowmick et al., 2020) that firstly extracts level-specific embeddings for each subgraph in the constructed hierarchy of subgraphs such that the final node representations are obtained by aggregating these level-specific embeddings.

A.3 PARAMETER SETTINGS

We set the common parameters of the random walk-based approaches, such as window size, the number of walks, and walk length to mostly preferred well-performing values 10, 80, and 10, respectively. The number of negative samples for NODE2VEC and LINE were chosen as 5 since they perform pretty well. We chose 0.01 for the damping parameter of LOUVAINNE and fixed the rank parameter of NETSMF to 256 and the number of rounds to $2,000$. We have chosen smaller values for the number of rounds on the large-scale networks because of the high memory demand. As suggested by the authors, we run the RANDNE approach with the adjacency matrix for $q = 2$ and $weights = [1, 1, 10^{-2}]$ and we performed the experiments with the recommended defaults values for the parameters of the PRONE algorithm. For the SH-LDM and SH-LDM-RE we optimize the negative log-likelihood via the Adam (Kingma & Ba, 2017) optimizer while setting the learning rate to $lr = 0.1$ for the all networks. We initialize the latent variables using the top-k eigenvectors of the adjacency matrix spectral decomposition (we set maximum $k = 10$, if the embedding dimension exceeds that threshold we use zeros for the rest dimensions which guarantees that the spectral proximity metric is preserved). In addition, we use the first 500 iterations to learn the scale of the initial latent positions and tune the bias terms. We terminate training after $15,000$ iterations of the algorithm. Lastly, we build the hierarchical structure by running the k-means procedure every $t = 25$ iterations and consider it fixed in any other case. The two proposed models make predictions directly based on the learned Poisson rates of the test pairs $\{i, j\}$ which utilize the Euclidean norm as $\lambda_{ij} = \exp(\gamma_i + \gamma_j - ||z_i - z_j||_2)$.

A.4 EXPERIMENTAL SETUP

For the experimental setup of the link prediction task, we follow the commonly applied strategy (Perozzi et al., 2014; Grover & Leskovec, 2016), and we remove half of the edges of a given network by keeping the residual network connected. Since this strategy is not feasible for large-scale networks, we hide 30% of the edges for these networks. The edges in the residual network form the positive samples for the training set, and the removed links are considered the positive instances for the testing set. We sample the same number of node pairs that are not the edges of the original network to construct the negative instances for the testing and training sets. We utilize the residual network to learn the node embeddings and design a feature vector for each node pair sample by applying a binary operator (Grover & Leskovec, 2016). The detailed list of the operators are given in Table 2

Table 2: Binary operators for constructing feature vectors for node pair samples. Each definition corresponds to d-th coordinate of the embedding vectors $\mathbf{z}_i$ and $\mathbf{z}_j$.

	Average	Hadamard	Weighted L_1	Weighted L_2
Operator	$(z_{i,d} + z_{j,d})/2$	$(z_{i,d} \times z_{j,d})$	$\lvert z_{i,d} - z_{j,d}\rvert$	$\lvert z_{i,d} - z_{j,d}\rvert^2$

A.4.1 A SPECTRAL CLUSTERING INITIALIZATION

A traditional initialization for LSMs is through the Multidimensional Scaling (MDS) algorithm, usually applied on the geodesic distance between the nodes of the network (Hoff et al., 2002; Krivitsky et al., 2009). Classical MDS is computationally expensive ($\mathcal{O}(N^3)$) and thus infeasible for large datasets. In addition, calculating the full geodesic distance matrix makes this initialization method prohibited even for computational friendly versions of the algorithm such as *split-and-combine* MDS (Tzeng et al., 2008) and Chalmer's Linear Iteration algorithm (Chalmers, 1996). As a scalable initialization for the latent variables we use the leaf centroid values of a spectral clustering procedure using hierarchical k-means over the first k generalized eigenvectors of an affinity matrix which was set as the adjacency matrix (Lei & Rinaldo, 2015), with other alternatives being any version of the Laplacian matrix (Jianbo Shi & Malik, 2000; Ng et al., 2001). Solving the eigenproblem for a few numbers of eigenvalues can be done efficiently through Lanczos method (Golub & Van Loan, 1996) due to the high sparsity of real large-scale networks. We use leaf centroid values rather than the corresponding eigenvectors since the spectral space does not represent homophily and transitivity in the same sense as the LDM. Therefore, we use leaf cluster values to provide an initial measurement of the Euclidean distance between fine groups of network nodes and allow the optimization to discover the enclosed structure and dynamics. Results using these initialization strategies are provided in the supplementary material.

A.5 HIERARCHICAL NETWORK VISUALIZATION

In order to discover the underlying structure for large networks, we use the learned embeddings of SH-LDM-RE and their corresponding cluster labels and re-order the adjacency matrix accordingly. The results for *YouTube*, *Flickr* and *Flixster* datasets for an embedding size of $D = 2$ and a tree height of $L = 3$ are shown in Figure 1. We witness that SH-LDM-RE successfully captures the network anatomy and can provide insights on how the structure is propagated across the different levels of resolution of the extracted network communities.

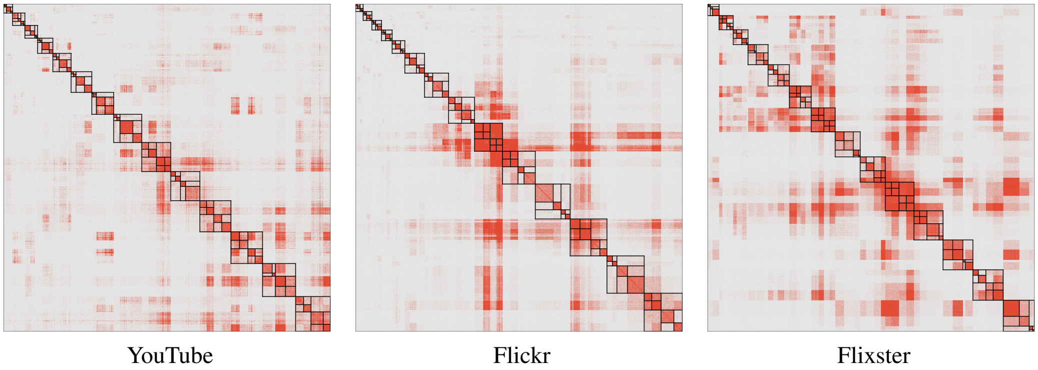

YouTube Flickr Flixster

Figure 1: Large-scale networks visualizations of the ordered adjacency matrices based on the learned $D = 2$ embeddings of SH-LDM-RE for a hierarchical structure of $L = 3$.

A.6 VISUALIZATION

In this part, we provide a validity test of the t-SNE constructed Space (t-SNES), for all baselines and proposed methods, as summarized by Figure 2. We provide the labeled-colored True Embedding Space (TES) for $D = 2$, as well as for $D = 2$ and $D = 128$ mapped to $D = 2$ via the use of t-SNE for *Cora* and *DBLP*, and report the performance over the tasks of network reconstruction, clustering and classification respectively reporting the AUC, the Normalized Mutual Information (NMI) and Micro-F_1 scores. As in the main paper we witness that our proposed frameworks SH-LDM and SH-LDM-RE are have the most consistent performance across the different downstream tasks.

A.7 CLASSIFICATION RESULTS

In Table 3 we provide the Macro-F_1 scores for the two large-scale and moderate-sized networks considered in the main paper. Again, for this metric our proposed methods outperform the baselines with LOUVAINNE having more or less on par performance. Despite the very good classification performance of LOUVAINNE, our method significantly outperforms the competitor on link prediction and thus is positioned as a stronger GRL approach.

A.8 TIME COMPLEXITY ANALYSIS

Here, we provide the theoretical complexity in big $\mathcal{O}$ notation for all of the considered models and our proposed frameworks. The results are given in Table A.8 and show that SH-LDM is a powerful representation learning tool that also scales nicely with the total size of the network. As we can see

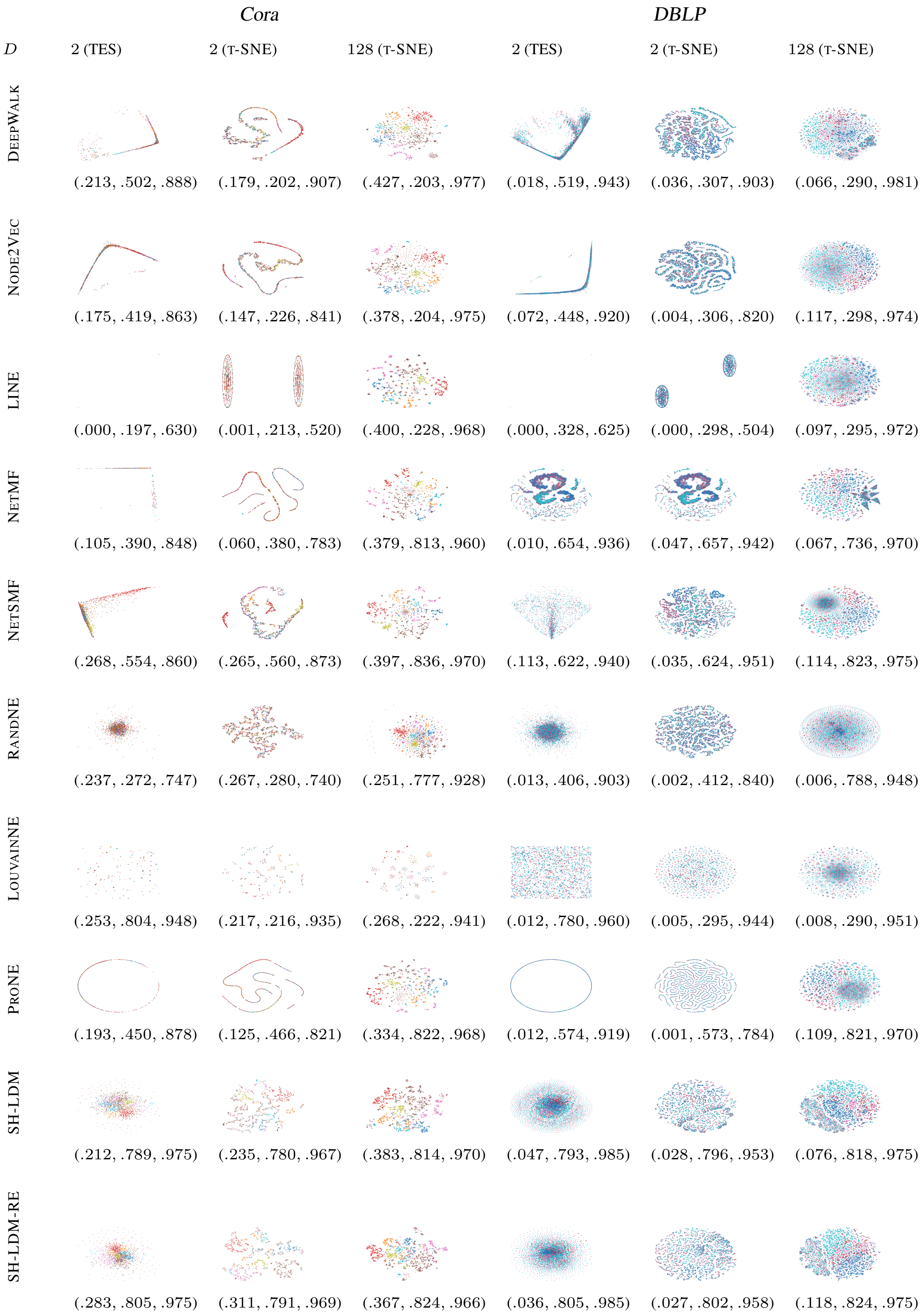


Figure 2: The first two columns for each dataset represent the node embeddings learned in two-dimensional space and T-SNE algorithm is applied for the second and the third columns to reduce the dimension size for the visualization task and to demonstrate the influence of T-SNE algorithm on the embeddings. For each network, NMI, Micro-F_1 and AUC scores are reported, respectively.

Table 3: Macro-F1 scores for varying representation sizes over two moderate-sized and two large-scale networks. The symbol ”-” indicates that the running time of the corresponding model takes more than 20 hours and ”x” shows that the method is not able to run due to insufficient memory space.

	Cora			*DBLP*			*Amazon*			*YouTube*		
DIMENSION	2	3	8	2	3	8	2	3	8	2	3	8
DEEPWALK	.474	.698	.806	.448	.546	.772	.181	.479	.849	.201	.258	.071
NODE2VEC	.368	.653	**.816**	.370	.470	.780	.069	.221	.801	-	-	-
LINE	.062	.142	.125	.172	.253	.244	.001	.001	.001	.059	.069	.071
NETMF	.350	.629	.785	.604	.660	.726	x	x	x	x	x	x
NETSMF	.534	.682	.802	.571	.692	.783	.280	.512	.734	.230	.271	.310
RANDNE	.213	.265	.458	.354	.426	.586	.204	.372	.697	.128	.135	.186
LOUVAINNE	.789	.792	.793	.757	.791	**.802**	**.920**	.922	**.924**	**.307**	**.306**	.306
PRONE	.413	.574	.756	.552	.609	.741	.401	.676	.873	.168	.195	.290
SH-LDM	.781	.790	.802	.773	.795	.746	.902	**.925**	.830	.238	.282	.335
SH-LDM-RE	**.790**	**.796**	.807	**.784**	**.803**	.795	.912	.900	.851	.242	.284	**.335**

from Figure 3 we see that essentially the complexity of SH-LDM grows linearly with the number of network edges and thus has very similar time complexity as the case-control variant.

Table 4: Complexity analysis of methods. V: vertex set, E: edge set, $\mathcal{N}$: number of walks, $\mathcal{L}$: walk length, H: height of the hierarchical tree, D: node representation size, k: number of negative instances, q: order value, c: Chebyshev expansion order, γ: window size, α_1 and α_2 constants such as $\alpha_1, \alpha_2 \ll |V|$.

Method	Complexity
DEEPWALK	$\mathcal{O}\left(\gamma\|V\|\log\left(\|V\|\right)\mathcal{N}\mathcal{L}\mathcal{D}\right)$
NODE2VEC	$\mathcal{O}\left(\gamma\|V\|\mathcal{N}\mathcal{L}\mathcal{D}k\right)$
LINE	$\mathcal{O}\left(\|E\|Dk\right)$
NETMF	$\mathcal{O}\left(\|V\|^2D\right)$
NETSMF	$\mathcal{O}\left(\|E\|(\gamma+D)+\|V\|D^2+D^3\right)$
RANDNE	$\mathcal{O}\left(\|V\|D^2+\|E\|Dq\right)$
LOUVAINNE	$\mathcal{O}\left(\|E\|\mathcal{H}+\|V\|D\right)$
PRONE	$\mathcal{O}\left(\|V\|D^2+\|E\|c\right)$
CC-LDM	$\mathcal{O}\left(\alpha_1\|E\|D\right)$
SH-LDM	$\mathcal{O}\left(\alpha_2\|V\|\log\|V\|D\right)$

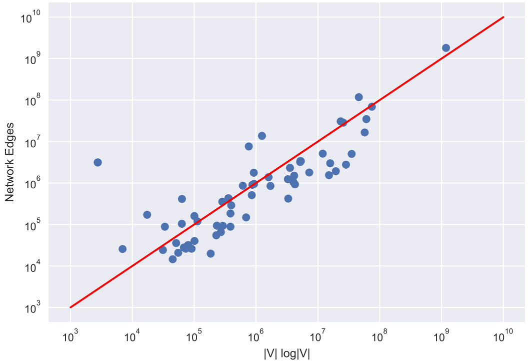


Figure 3: Scatter plot on a log-log scale for the number of network edges versus $|V|\log|V|$ with $|V|$ the vertex set, for 63 datasets of the SNAP library (Leskovec & Krevl, 2014).

A.9 BIPARTITE MODELING

In the main paper, our study concentrates on modeling undirected networks. Nevertheless, SH-LDM and SH-LDM-RE generalize to both directed and bipartite graphs. In the following, we provide the mathematical extension for the bipartite case (the directed network formulation of our proposed model can be considered a special case of the bipartite framework in which self-links are removed and thus omitted from the below log-likelihood). We further present a case study based on the *Filmtrust* bipartite network (Guo et al., 2013) which consists of an adjacency matrix of size 1505×2069. For a bipartite network we can formulate the log-likelihood of the Poisson linear model as:

$$\log P(Y|\boldsymbol{\lambda}) = \sum_{i,j} \Big(y_{ij} \log(\lambda_{ij}) - \lambda_{ij} \Big) = \sum_{i,j:y_{ij}=1} \log(\lambda_{ij}) \; - \; \sum_{i,j} \lambda_{ij} \; , \tag{1}$$

where we have disregarded the term $-\log{(y_{ij}!)}$ which is equal to zero for binary networks. Based on the LDM we can then define the Poisson rate as:

$$\lambda_{ij} = \exp\Big(\psi_i + \omega_j - d\big(\mathbf{w}_i, \mathbf{v}_j\big)\Big), \tag{2}$$

where ψ_i and ω_j are the corresponding random effects and $\{\mathbf{w}_i, \, \mathbf{v}_j\}$ the latent variables of the two disjoint sets of network nodes, N_1 and N_2 respectively. We note that in Eq. equation 2 we can also include potential covariate information of the network which can be absorbed in the random effects. Following the same format as of the main paper, we can define a scalable hierarchical version of a bipartite LDM based on a clustering procedure while again using a multiresolution KD tree. In this setting, we use our divisive Euclidean distance hierarchical clustering procedure over the concatenation $\mathbf{z} = [\mathbf{w}; \mathbf{v}]$ of the two sets of latent variables. We can then mimic the steps described in the methods section of the main paper applied to a non-symmetric case and approximate the pairwise distance matrix $D_{N_1 \times N_2}$ in the same manner. In this formulation, a centroid is considered a leaf if the corresponding tree-cluster contains less than $\log(N1)$ of the latent variables $\mathbf{w}$ or less than $\log(N_2)$ of the latent variables $\mathbf{v}$. We thereby define the Scalable Hierarchical-Latent Distance Model for a bipartite network as:

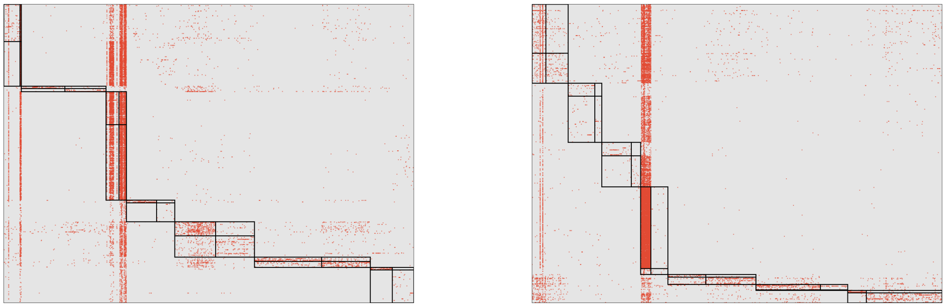
$$\begin{aligned}\log P(Y|\boldsymbol{\lambda}) \quad &= \textstyle\sum_{y_{i,j}=1} \left(\psi_i + \omega_j - ||\mathbf{w}_i - \mathbf{v}_j||_2 \right) - \sum_{k_L}^{K_L} \left(\sum_{i,j \in C_{k_L}} e^{(\psi_i + \omega_j - ||\mathbf{w}_i - \mathbf{v}_j||_2)} \right) &(3)\\ &\quad - \textstyle\sum_{l}^{L} \left(\sum_{k_l, k_l'} e^{-||\boldsymbol{\mu}_{k_l} - \boldsymbol{\mu}_{k_l'}||_2} \left(\sum_{i \in C_{k_l}} e^{\psi_i} \right) \left(\sum_{j \in C_{k_l'}} e^{\omega_j} \right) \right), &(4)\end{aligned}$$

where $\boldsymbol{\mu}_k$ are the latent centroids which have absorbed the dependency of both sets of latent variables $\{\mathbf{w}_i, \mathbf{v}_j\}$.

We present the results of a case-study where we used the bipartite version of both SH-LDM and SH-LDM-RE to analyze *Filmtrust*. In addition, we evaluate the predictive power of our model by zeroing out 50% of the network edges which we then try to predict, as also described in the main paper. Since the proposed spectral clustering initialization method does not generalize to the bipartite scenario, we initialized randomly both sets of latent variables and leave more accurate initialization frameworks as future work. The predictive performance is summarized in Table 5, where we see that both SH-LDM and SH-LDM-RE are able to successfully infer missing links but with a clear performance gap between them, highlighting the importance of random-effects also when modeling bipartite networks. Furthermore, it is evident from Figure 4 that both SH-LDM and SH-LDM-RE can discover and account for the network structure. Lastly, we provide visualizations of the constructed latent space of SH-LDM and SH-LDM-RE as seen in Figure 4.

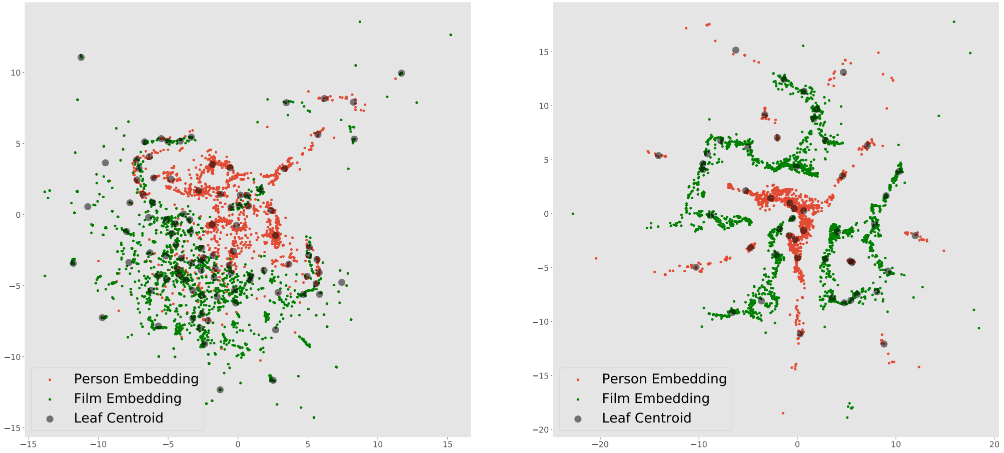
Table 5: Predictive performance on the *Filmtrust* network for embedding dimensionality $D = 2$.

	AUC-ROC	AUC-PR
SH-LDM-RE	0.967	0.967
SH-LDM	0.899	0.918



SH-LDM-RE SH-LDM

Figure 4: *Filmtrust* network visualizations of the ordered adjacency matrix based on the learned $D = 2$ embeddings for a hierarchical structure visualized for $L = 2$.



SH-LDM-RE SH-LDM

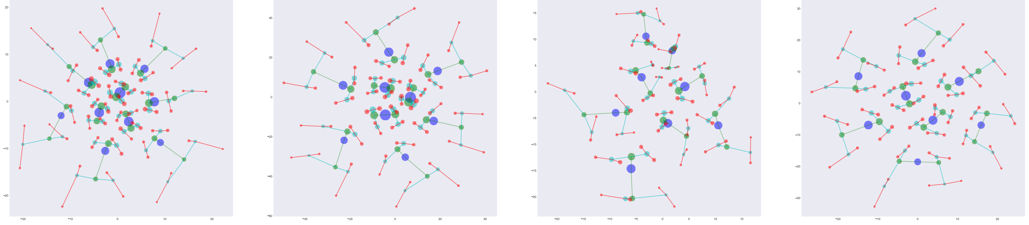
Figure 5: Constructed latent space based on the learned $D = 2$ embeddings and the leaf centroids of the constructed hierarchical structure for *Filmtrust*.

A.10 HIERARCHICAL EMBEDDINGS AND ADJACENCY MATRIX VISUALIZATIONS

In this section we provide additional visualizations for the networks studied throughout this paper, for both the case of moderate-sized and large-scale networks.

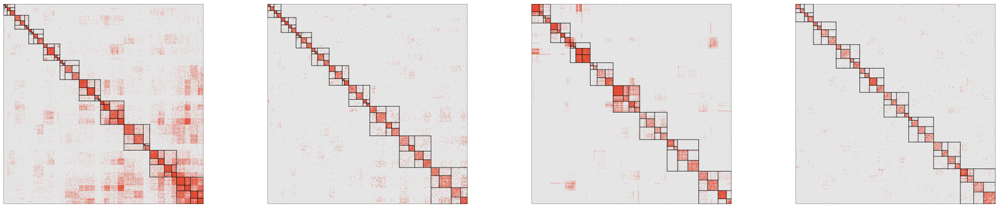
A.10.1 MODERATE-SIZED NETWORKS

Through the proposed SH-LDM-RE framework we are able to visualize the network structure in a hierarchical manner. We provide the visualization of the hierarchical centroids for moderate-sized networks in Figure 6 where we observe how the tree evolves as we move down the hierarchy expressing with increasing detail the network anatomy. Moreover, we provide the ordered adjacency matrices for the moderate-sized networks for both SH-LDM and SH-LDM-RE in Figures 7 and 8. Based on these figures we can observe that both versions of the model can successfully discover the network structure.



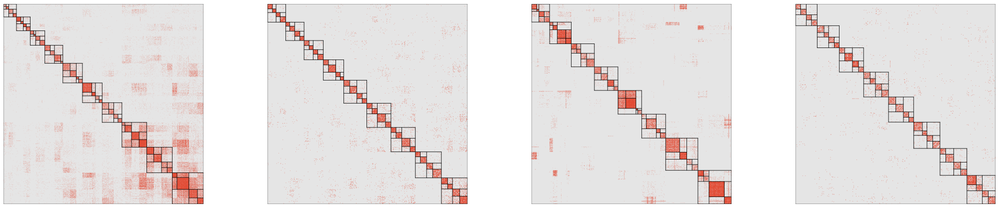
AstroPh *HepTh* Facebook GrQc

Figure 6: Moderate-sized networks visualizations of the obtained centroids based on the hierarchical k-means clustering of the learned $D = 2$ embeddings for SH-LDM-RE for a hierarchical structure of $L = 4$. Each different centroid color represents the tree level which with blue we denote $L = 1$, green $L = 2$, cyan $L = 3$ and red $L = 4$. Each colored line denotes in which clusters the parent cluster splits across the network hierarchy. The size of each centroid is proportional on the number of network nodes the corresponding cluster contains.



AstroPh *HepTh* Facebook GrQc

Figure 7: Ordered adjacency matrices for SH-LDM-RE.

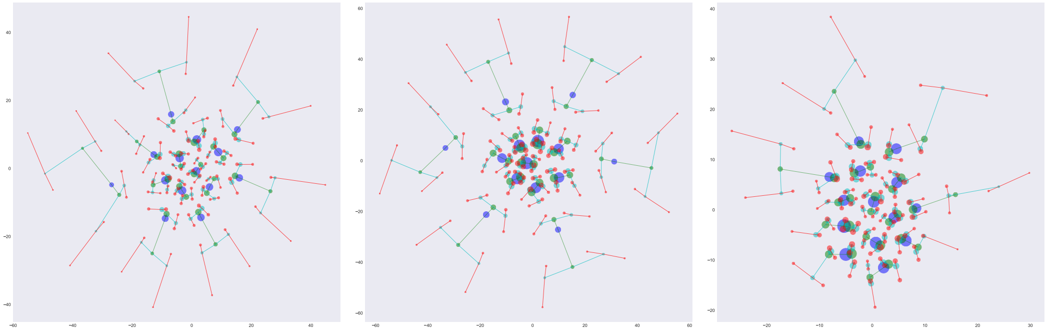


AstroPh *HepTh* Facebook GrQc

Figure 8: Ordered adjacency matrices for SH-LDM.

A.10.2 LARGE-SCALE NETWORKS

The visualizations of the hierarchical centroids for the large-scale networks is given in Figure 9 where again we observe how the network can be expressed in each level with an increasing amount of detail and characteristics. The proposed framework shows promising results as a large-scale network compression tool which is able to maintain high-order information with very few memory requirements.



YouTube Flickr Flixster

Figure 9: Large-scale networks visualizations of the obtained centroids based on the hierarchical k-means clustering of the learned $D = 2$ embeddings for SH-LDM-RE for a hierarchical structure of $L = 4$. Each different centroid color represents the tree level which with blue we denote $L = 1$, green $L = 2$, cyan $L = 3$ and red $L = 4$. Each colored line denotes in which clusters the parent cluster splits across the network hierarchy. The size of each centroid is proportional to the number of network nodes the corresponding cluster contains.

A.11 PREDICTIVE PERFORMANCE ANALYTICAL RESULTS

We here provide results describing the predictive performance of the proposed SH-LDM-RE, SH-LDM and the different baselines including a direct comparison with the Case-Control Latent Distance Model (CC-LDM) and with random effects (CC-LDM-RE). We outline the predictive capabilities via the AUC-ROC and AUC-PR metrics. We provide error bars based on an average of three runs for both our proposed framework and the case-control for the moderate-sized networks. For the large-scale networks due to time constraints we only provide error bars for our proposed method and for $D = 2, 3$. In addition, we were unable to run the case-control variant of the LDM for $D = 16$ due to our hardware memory limitations. In the main paper, the results are based on a single run for all models which is found to be almost identical to the mean value of the different runs.

A.11.1 AUC-ROC METRIC

We summarize the AUC-ROC performance metric for the different models while predicting on the moderate-sized networks in Tables 6 to 9. The obtained results verify that the family of LDMs such as SH-LDM and CC-LDM outperform the rest of the baselines for small dimensions while being robust as the dimensionality increases and where also the baselines become more competitive. Both SH-LDM-RE and CC-LDM-RE outline the importance of random effects when modeling the different networks. Comparing our proposed method with the case-control version we observe that there are occasions where both models outperform each other. The gap though is quite more often for the case of SH-LDM-RE and SH-LDM which usually outperforms the corresponding case-control version. The predictive performance for the large-scale networks is summarized by Tables 10 to 12, where again we observe that the LDMs outperform the baselines and our proposed method is again on par with the case-control version. What is more, NETSMF provides quite a competitive performance equivalent to the LDM family. It is important to note though that both the scalable NETSMF version as well as the full version NETMF are drastically outperformed in the moderate-sized networks showing that these models are not robust across different network sizes while our proposed method as well as the case-control variant are.

Table 6: AUC-ROC scores for varying dimension sizes on the *AstroPh* network.

	$D=2$	$D=3$	$D=8$	$D=32$	$D=64$	$D=128$
DeepWalk	0.831	0.883	0.945	0.953	0.955	0.961
Node2Vec	0.825	0.886	0.951	0.957	0.940	0.902
LINE	0.632	0.832	0.902	0.948	0.955	0.960
NetMF	0.800	0.805	0.814	0.839	0.853	0.868
NetSMF	0.828	0.854	0.891	0.919	0.936	0.943
RandNE	0.524	0.545	0.554	0.618	0.642	0.681
LouvainNE	0.798	0.805	0.813	0.819	0.840	0.864
ProNE	0.768	0.843	0.907	0.947	0.957	0.965
CC-LDM	0.906±0.004	0.925±0.001	0.968±0.001	0.978±0.001	0.976±0.001	0.977±0.001
CC-LDM-RE	0.932±0.002	0.938±0.009	0.974±0.007	0.979±0.000	0.980±0.000	0.979±0.000
Fr-LSM	0.917±0.003	0.940±0.001	0.960±0.000	0.961±0.000	0.961±0.001	0.961±0.000
Fr-LSM-RE	0.931±0.001	0.948±0.001	0.962±0.001	0.964±0.001	0.964±0.002	0.965±0.001

Table 7: AUC-ROC scores for varying dimension sizes on the *GrQc* network.

	$D=2$	$D=3$	$D=8$	$D=32$	$D=64$	$D=128$
DeepWalk	0.845	0.886	0.919	0.930	0.936	0.937
Node2Vec	0.809	0.852	0.884	0.894	0.897	0.895
LINE	0.688	0.867	0.920	0.919	0.918	0.910
NetMF	0.830	0.841	0.860	0.875	0.891	0.916
NetSMF	0.756	0.772	0.805	0.847	0.863	0.883
RandNE	0.534	0.551	0.560	0.604	0.630	0.660
LouvainNE	0.861	0.866	0.868	0.873	0.883	0.898
ProNE	0.818	0.853	0.883	0.915	0.918	0.909
CC-LDM	0.880±0.002	0.888±0.008	0.931± 0.002	0.943±0.001	0.942±0.002	0.940±0.002
CC-LDM-RE	0.908±0.000	0.918±0.003	0.945±0.001	0.949±0.003	0.945±0.003	0.950±0.001
Sh-LDM	0.898±0.003	0.916±0.002	0.944± 0.000	0.946±0.001	0.946±0.001	0.947±0.001
Sh-LDM-Re	0.910±0.001	0.929± 0.002	0.953±0.004	0.953±0.000	0.952±0.000	0.953±0.001

Table 8: AUC-ROC scores for varying dimension sizes on the *Facebook* network.

	$D=2$	$D=3$	$D=8$	$D=32$	$D=64$	$D=128$
DeepWalk	0.958	0.978	0.986	0.985	0.984	0.983
Node2Vec	0.914	0.960	0.988	0.987	0.987	0.988
LINE	0.751	0.958	0.979	0.983	0.982	0.979
NetMF	0.872	0.872	0.936	0.971	0.973	0.974
NetSMF	0.907	0.919	0.976	0.986	0.987	0.987
RandNE	0.614	0.606	0.657	0.726	0.789	0.815
LouvainNE	0.957	0.951	0.958	0.963	0.971	0.973
ProNE	0.900	0.901	0.971	0.983	0.985	0.984
CC-LDM	0.982±0.000	0.989±0.000	0.992±0.000	0.992±0.000	0.992±0.000	0.992±0.000
CC-LDM-RE	0.989±0.001	0.993±0.000	0.994±0.000	0.993±0.000	0.993±0.000	0.993±0.000
Sh-LDM	0.980±0.002	0.986±0.000	0.986±0.002	0.987±0.000	0.986±0.000	0.985±0.000
Sh-LDM-Re	0.986±0.000	0.990±0.000	0.988±0.002	0.989±0.000	0.989±0.000	0.989± 0.000

Table 9: AUC-ROC scores for varying dimension sizes on the *HepTh* network.

	$D = 2$	$D = 3$	$D = 8$	$D = 32$	$D = 64$	$D = 128$
DeepWalk	0.773	0.840	0.874	0.874	0.889	0.896
Node2Vec	0.780	0.824	0.881	0.871	0.858	0.848
LINE	0.659	0.804	0.874	0.873	0.858	0.857
NetMF	0.757	0.765	0.792	0.819	0.833	0.858
NetSMF	0.705	0.730	0.801	0.836	0.836	0.858
RandNE	0.519	0.499	0.509	0.531	0.554	0.569
LouvainNE	0.774	0.840	0.874	0.874	0.889	0.897
ProNE	0.678	0.705	0.827	0.860	0.872	0.866
CC-LDM	0.799±0.017	0.821±0.009	0.881±0.003	0.904±0.006	0.909±0.005	0.905±0.006
CC-LDM-RE	0.827 ±0.011	0.846±0.005	0.903±0.004	0.916±0.002	0.913±0.002	0.915±0.003
Sh-LDM	0.847±0.009	0.863±0.003	0.912±0.001	0.914±0.002	0.915±0.001	0.914±0.002
Sh-LDM-Re	0.869±0.005	0.882±0.003	0.923±0.0004	0.926±0.001	0.927±0.001	0.926±0.002

Table 10: AUC-ROC scores for varying dimension sizes on the *Youtube* network.

	$D = 2$	$D = 3$	$D = 8$	$D = 16$
DeepWalk	0.822	0.891	0.921	0.923
Node2Vec	-	-	-	-
LINE	0.660	0.832	0.878	0.890
NetMF	x	x	x	x
NetSMF	0.939	0.940	0.949	0.953
RandNE	0.672	0.700	0.762	0.801
LouvainNE	0.820	0.819	0.815	0.829
ProNE	0.691	0.761	0.861	0.879
CC-LDM	0.898	0.902	0.952	*
CC-LDM-RE	0.939	0.946	0.952	*
Sh-LDM	0.899±0.002	0.919±0.000	0.935	0.936
Sh-LDM-Re	0.950±0.001	0.949±0.001	0.957	0.957

Table 11: AUC-ROC scores for varying dimension sizes on the *Flickr* network.

	$D = 2$	$D = 3$	$D = 8$	$D = 16$
DeepWalk	0.889	0.937	0.972	0.977
Node2Vec	-	-	-	-
LINE	0.685	0.889	0.921	0.925
NetMF	x	x	x	x
NetSMF	0.974	0.977	0.980	0.983
RandNE	0.833	0.869	0.903	0.916
LouvainNE	0.898	0.899	0.909	0.922
ProNE	0.623	0.819	0.908	0.927
CC-LDM	0.972	0.979	0.990	*
CC-LDM-RE	0.983	0.983	0.990	*
Sh-LDM	0.972±0.000	0.979±0.000	0.986	0.983
Sh-LDM-Re	0.981±0.001	0.985±0.001	0.988	0.988

Table 12: AUC-ROC scores for varying dimension sizes on the *Flixster* network.

	$D = 2$	$D = 3$	$D = 8$	$D = 16$
DeepWalk	0.820	0.866	0.921	0.932
Node2Vec	-	-	-	-
LINE	0.523	0.868	0.936	0.978
NetMF	x	x	x	x
NetSMF	0.987	0.987	0.987	0.988
RandNE	0.700	0.739	0.835	0.886
LouvainNE	0.735	0.718	0.746	0.726
ProNE	0.756	0.803	0.846	0.868
CC-LDM	0.882	0.919	0.955	*
CC-LDM-RE	0.962	0.964	0.967	*
Sh-LDM	0.898±0.001	0.917±0.001	0.932	0.933
Sh-LDM-Re	0.962±0.002	0.965±0.005	0.971	0.972

A.11.2 AUC-PR Metric

We summarize the AUC-PR performance metric for the different models while predicting on the moderate-sized networks in Tables 13 to 16. The obtained results again verify that the family of LDMs outperforms the baselines, especially for low-dimensions. The large-scale network performance is summarized by Tables 18 to 17 and verify the surplus of the expressive power LDMs have. If we compare again our proposed framework with the case-control variant we see that indeed these methods are on par.

Table 13: AUC-PR scores for varying dimension sizes on the *AstroPh* network.

	$D = 2$	$D = 3$	$D = 8$	$D = 32$	$D = 64$	$D = 128$
DeepWalk	0.791	0.868	0.952	0.963	0.965	0.969
Node2Vec	0.787	0.875	0.955	0.966	0.952	0.920
LINE	0.757	0.778	0.916	0.959	0.964	0.970
NetMF	0.789	0.798	0.807	0.829	0.841	0.859
NetSMF	0.667	0.765	0.874	0.951	0.965	0.972
RandNE	0.550	0.564	0.576	0.645	0.675	0.713
LouvainNE	0.854	0.857	0.861	0.865	0.877	0.891
ProNE	0.755	0.835	0.914	0.958	0.972	0.979
CC-LDM	0.923±0.003	0.940±0.000	0.974±0.000	0.981±0.000	0.980±0.001	0.981±0.001
CC-LDM-RE	0.945± 0.001	0.951±0.006	0.978±0.001	0.982± 0.001	0.983±0.000	0.983±0.001
Sh-LDM	0.930±0.001	0.949± 0.001	0.967±0.001	0.968±0.001	0.968±0.000	0.968±0.000
Sh-LDM-Re	0.944±0.001	0.957±0.001	0.972 ±0.001	0.971±0.001	0.971±0.000	0.972±0.000

Table 14: AUC-PR scores for varying dimension sizes on the *GrQc* network.

	$D = 2$	$D = 3$	$D = 8$	$D = 32$	$D = 64$	$D = 128$
DeepWalk	0.834	0.900	0.939	0.948	0.955	0.955
Node2Vec	0.748	0.863	0.927	0.929	0.931	0.931
LINE	0.791	0.852	0.941	0.945	0.944	0.941
NetMF	0.862	0.872	0.887	0.903	0.919	0.940
NetSMF	0.729	0.769	0.847	0.913	0.933	0.947
RandNE	0.541	0.554	0.579	0.643	0.676	0.704
LouvainNE	0.904	0.905	0.906	0.910	0.915	0.923
ProNE	0.832	0.868	0.908	0.940	0.940	0.935
CC-LDM	0.912±0.002	0.917±0.005	0.949±0.001	0.958±0.000	0.957±0.001	0.956±0.001
CC-LDM-RE	0.932±0.002	0.939±0.002	0.960±0.001	0.962±0.001	0.960±0.001	0.963±0.000
Sh-LDM	0.924±0.001	0.939±0.001	0.959±0.000	0.961±0.001	0.960±0.000	0.961±0.001
Sh-LDM-Re	0.935±0.001	0.946±0.001	0.963±0.001	0.965±0.001	0.964±0.000	0.965±0.000

Table 15: AUC-PR scores for varying dimension sizes on the *Facebook* network.

	$D = 2$	$D = 3$	$D = 8$	$D = 32$	$D = 64$	$D = 128$
DeepWalk	0.948	0.971	0.983	0.985	0.985	0.984
Node2Vec	0.886	0.947	0.984	0.983	0.983	0.984
LINE	0.833	0.953	0.973	0.982	0.979	0.972
NetMF	0.882	0.881	0.930	0.965	0.964	0.967
NetSMF	0.905	0.921	0.971	0.983	0.985	0.985
RandNE	0.667	0.656	0.708	0.772	0.833	0.855
LouvainNE	0.949	0.946	0.947	0.949	0.956	0.958
ProNE	0.863	0.901	0.966	0.985	0.989	0.990
CC-LDM	0.983 ± 0.000	0.988 ±0.000	0.909 ±0.000	0.990 ±0.000	0.990 ±0.001	0.990 ±0.001
CC-LDM-RE	0.990 ± 0.001	0.992 ±0.000	0.993 ±0.000	0.992 ± 0.000	0.992 ±0.000	0.992 ±0.000
Sh-LDM	0.978 ±0.001	0.985 ± 0.000	0.986 ±0.000	0.985 ±0.000	0.985 ±0.000	0.984 ±0.000
Sh-LDM-Re	0.987 ±0.000	0.989 ±0.000	0.989 ±0.001	0.989 ±0.000	0.989 ±0.000	0.989±0.000

Table 16: AUC-PR scores for varying dimension sizes on the *HepTh* network.

	$D = 2$	$D = 3$	$D = 8$	$D = 32$	$D = 64$	$D = 128$
DeepWalk	0.765	0.846	0.896	0.896	0.914	0.923
Node2Vec	0.759	0.835	0.910	0.911	0.903	0.894
LINE	0.766	0.803	0.900	0.908	0.900	0.898
NetMF	0.778	0.790	0.817	0.847	0.863	0.886
NetSMF	0.733	0.755	0.830	0.876	0.885	0.903
RandNE	0.527	0.498	0.511	0.534	0.559	0.580
LouvainNE	0.818	0.824	0.827	0.840	0.846	0.851
ProNE	0.695	0.726	0.858	0.895	0.903	0.902
CC-LDM	0.838 ± 0.015	0.857 ±0.007	0.907 ±0.003	0.928 ±0.004	0.931 ±0.003	0.928 ±0.004
CC-LDM-RE	0.864 ± 0.009	0.879 ±0.003	0.925 ±0.002	0.936 ± 0.001	0.934 ±0.001	0.935 ±0.001
Sh-LDM	0.869 ±0.006	0.891 ± 0.002	0.931 ±0.001	0.934 ±0.001	0.934 ±0.001	0.934 ±0.001
Sh-LDM-Re	0.891 ±0.001	0.908 ±0.002	0.940 ±0.001	0.943 ±0.001	0.943 ±0.001	0.944±0.001

Table 17: AUC-PR scores for varying dimension sizes on the *Youtube* network.

	$D = 2$	$D = 3$	$D = 8$	$D = 16$
DeepWalk	0.804	0.880	0.934	0.939
Node2Vec	-	-	-	-
LINE	0.766	0.818	0.882	0.905
NetMF	x	x	x	x
NetSMF	0.954	0.955	0.962	0.964
RandNE	0.724	0.751	0.810	0.845
LouvainNE	0.859	0.859	0.859	0.865
ProNE	0.641	0.746	0.845	0.892
CC-LDM	0.916	0.922	0.960	*
CC-LDM-RE	0.955	0.959	0.963	*
Sh-LDM	0.914±0.001	0.933±0.001	0.951	0.950
Sh-LDM-Re	0.954±0.001	0.960±0.001	0.967	0.969

Table 18: AUC-PR scores for varying dimension sizes on the *Flickr* network.

	$D = 2$	$D = 3$	$D = 8$	$D = 16$
DeepWalk	0.836	0.923	0.975	0.981
Node2Vec	-	-	-	-
LINE	0.694	0.902	0.969	0.976
NetMF	x	x	x	x
NetSMF	0.980	0.983	0.984	0.986
RandNE	0.866	0.895	0.920	0.929
LouvainNE	0.917	0.913	0.920	0.928
ProNE	0.722	0.715	0.875	0.961
CC-LDM	0.978	0.982	0.992	*
CC-LDM-RE	0.987	0.991	0.992	*
Sh-LDM	0.977±0.000	0.983±0.000	0.990	0.9831
Sh-LDM-Re	0.988±0.000	0.990±0.000	0.993	0.991

Table 19: AUC-PR scores for varying dimension sizes on the *Flixster* network.

	$D = 2$	$D = 3$	$D = 8$	$D = 16$
DeepWalk	0.776	0.845	0.925	0.943
Node2Vec	-	-	-	-
LINE	0.638	0.873	0.932	0.968
NetMF	x	x	x	x
NetSMF	0.987	0.987	0.988	0.988
RandNE	0.739	0.777	0.864	0.907
LouvainNE	0.786	0.779	0.792	0.783
ProNE	0.700	0.784	0.843	0.878
CC-LDM	0.896	0.927	0.960	*
CC-LDM-RE	0.970	0.971	0.973	*
Sh-LDM	0.903±0.000	0.924±0.001	0.953	0.948
Sh-LDM-Re	0.969±0.001	0.974 ±0.005	0.976	0.979

A.12 QUALITY OF APPROXIMATION

We provide the negative log-likelihood comparison of SH-LDM and LDM where we input the solution of SH-LDM to the full LDM. The results are summarized in Figures 10 to 12 where we observe that the SH-LDM likelihood approximation corresponds well with the true full likelihood-providing systematically slightly lower likelihood estimates which we attribute to the SH-LDM inference minimizing the hierarchical approximation.

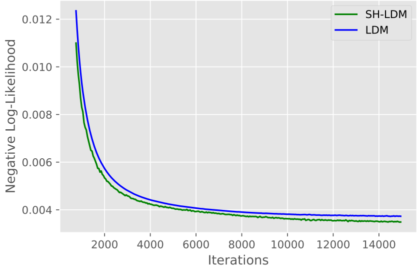


Figure 10: Negative log likelihood comparison between SH-LDM and LDM for *AstroPh* datasets with D = 2.

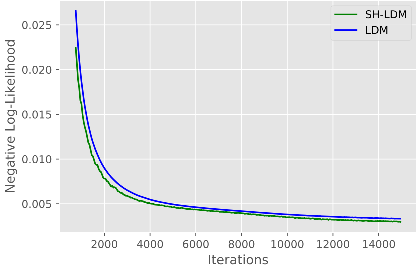


Figure 11: Negative log likelihood comparison between SH-LDM and LDM for *GrQc* dataset with D = 2.

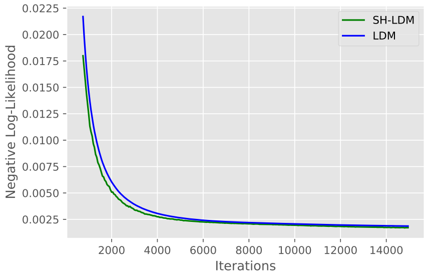


Figure 12: Negative log likelihood comparison between SH-LDM and LDM for *HepTh* dataset with D = 2.

A.12.1 INITIALIZATION PROCEDURE RESULTS

The configuration of the initial latent positions of the network nodes plays a vital role in the speed of convergence and the quality of the obtained solution. A popular and successful initialization schema for Latent Distance Models has been the Multidimensional Scaling (MDS) algorithm, usually applied on the geodesic distance between the nodes of the network (Hoff et al., 2002; **?**). Classical MDS is computational expensive, returning a complexity of $\mathcal{O}(N^3)$ and as result not scaling for large datasets. In addition, calculating the geodesic distance matrix (All-Pairs Shortest Paths) makes this initialization method prohibited even for computational friendly versions of the algorithm such as *split-and-combine* MDS (Tzeng et al., 2008) and Chalmer's Linear Iteration algorithm (Chalmers, 1996).

In this paper, we adopt a method for initializing latent space models which respects the computational constraints both in time and space that the analysis of large network enforces. Our approach is based on spectral clustering while using the hierarchical k-means procedure introduced previously. We perform spectral clustering over the first k generalized eigenvectors of the affinity matrix. Solving the eigenproblem for a few number of eigenvalues can be done efficiently through Lanczos method (Golub & Van Loan, 1996), due to the high sparsity of real large-scale networks. The clustering phase of the corresponding eigenvectors scales well through the use of the divisive Euclidean distance hierarchical clustering algorithm.

We initialize the latent positions of the network with their assigned leaf node centroid, obtained by the divisive Euclidean distance hierarchical clustering procedure. The reason behind using the centroid values rather than the corresponding eigenvector values, is due to the fact that the spectral space does not represent homophily and transitivity as in the same sense of the latent space model. Instead, we use the centroid values to provide an initial measurement about the distance between group of network nodes and then letting the network to discover the enclosed group structure and dynamics. Various choices exist for the affinity matrix leading to different spectral clustering algorithms. The most popular choices being the two normalized spectral clustering algorithms of Jianbo Shi & Malik (2000) and Ng et al. (2001). One potential problem using the Laplacian as the affinity matrix, is the multiplicity of the zero eigenvalue and the fact that numerical eigensolvers do not certainly converge to the true corresponding eigenvectors' values as analyzed in von Luxburg (2007). Despite that, the author concludes that this phenomenon does not affect dramatically the quality of clustering. A second problem, is that computing the k smallest eigenvalues of the affinity matrix is general more complex than calculating the largest. This can be an issue for the case of large scale networks. For this reason, we perform our spectral clustering directly over the adjacency matrix which requires the calculation of the largest eigenvalues and has been studied thoroughly in Lei & Rinaldo (2015).

In order to verify the success of our initialization procedure we predict the missing links for all networks based on the latent variables obtained by the different frameworks. More specifically we consider the spectral clustering algorithm of Ng et al. (2001) as $(\mathrm{SC\text{-}L})$, the direct spectral decomposition of the adjacency matrix as $(\mathrm{SC\text{-}A})$ and the classical mutlidimensional scaling algorithm as (MDS). We provide the AUC-ROC results for all the moderate-sized networks and one large-scale network in Tables 22 to 24. Where for the smaller networks such as *GrQc* and *Facebook* we observe that MDS and SC-L have very similar predictive power and both outperform SC-A. As we increase the size of the networks (where classical MDS is infeasible) we observe that SC-A completely outperforms SC-L due to the fact that the numerical eigensolvers of the Laplacian do not converge to the true corresponding eigenvectors and more attention has to be paid on how we decompose the Laplacian. This is not the case SC-A since obtaining the largest eigenvalues of the sparse adjacency matrix seems to be robust.

Table 20: AUC-ROC scores for *AstroPh*

Dimension	$D = 2$	$D = 3$	$D = 8$	$D = 10$
MDS	-	-	-	-
SC-A	0.7881 (0.0009)	0.8010 (0.0004)	0.8392 (0.0001)	0.8359 (0.0006)
SC-L	0.5693 (0.0067)	0.5653 (0.0042)	0.6010 (0.0007)	0.5979 (0.0029)

Table 21: AUC-ROC scores for *GrQc*

Dimension	$D = 2$	$D = 3$	$D = 8$	$D = 10$
MDS	0.8668 (0.0017)	0.8891 (0.0066)	0.9254 (0.0023)	0.9302 (0.0003)
SC-A	0.7957 (0.0004)	0.7990 (0.0021)	0.8193 (0.0036)	0.8181 (0.0046)
SC-L	0.8646 (0.0002)	0.8716 (0.0009)	0.9014 (0.0001)	0.9074 (0.0001)

Table 22: AUC-ROC scores for *Facebook*

Dimension	$D = 2$	$D = 3$	$D = 8$	$D = 10$
MDS	0.8989 (0.0124)	0.9065 (0.0070)	0.9535 (0.0001)	0.9557 (0.0002)
SC-A	0.8727 (0.0005)	0.9230 (0.0001)	0.9459 (0.0005)	0.9279 (0.0021)
SC-L	0.9537 (0.0013)	0.9523 (0.0019)	0.9657 (0.0008)	0.9670 (0.0029)

Table 23: AUC-ROC scores for *HepTh*

Dimension	$D = 2$	$D = 3$	$D = 8$	$D = 10$
MDS	-	-	-	-
SC-A	0.7467 (0.0004)	0.7530 (0.0011)	0.7783 (0.0009)	0.7806 (0.0013)
SC-L	0.7328 (0.0201)	0.7427 (0.0139)	0.7736 (0.0011)	0.7794 (0.0016)

Table 24: AUC-ROC scores for *YouTube*

Dimension	$D = 2$	$D = 3$	$D = 8$	$D = 10$
MDS	-	-	-	-
SC-A	0.5617 (0.0002)	0.5129 (0.0003)	0.7148 (0.0003)	0.7315 (0.0001)
SC-L	0.5036 (0.0007)	0.5029 (0.0004)	0.5028(0.0008)	0.5020 (0.0004)

REFERENCES

Ayan Kumar Bhowmick, Koushik Meneni, Maximilien Danisch, Jean-Loup Guillaume, and Bivas Mitra. LouvainNE: Hierarchical louvain method for high quality and scalable network embedding. In *WSDM*, pp. 43–51, 2020.

Matthew Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *Proceedings of the 7th Conference on Visualization '96*, VIS '96, pp. 127–ff., Washington, DC, USA, 1996. IEEE Computer Society Press. ISBN 0897918649.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, USA, 1996. ISBN 0801854148.

Aditya Grover and Jure Leskovec. Node2Vec: Scalable feature learning for networks. In *KDD*, pp. 855–864, 2016.

G. Guo, J. Zhang, and N. Yorke-Smith. A novel bayesian similarity measure for recommender systems. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2619–2625, 2013.

Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002. doi: 10.1198/016214502388618906.

Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. doi: 10.1109/34.868688.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Pavel N. Krivitsky, Mark S. Handcock, Adrian E. Raftery, and Peter D. Hoff. Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, 31(3):204 – 213, 2009.

Jing Lei and Alessandro Rinaldo. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1), Feb 2015. doi: 10.1214/14-aos1274.

Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection, June 2014.

Jure Leskovec and Julian J. Mcauley. Learning to discover social circles in ego networks. pp. 539–547, 2012.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1), 2007.

Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC'07)*, San Diego, CA, October 2007.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pp. 849–856, Cambridge, MA, USA, 2001. MIT Press.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online learning of social representations. *CoRR*, abs/1403.6652, 2014.

Bryan Perozzi, Vivek Kulkarni, Haochen Chen, and Steven Skiena. Don't walk, skip! online learning of multi-scale network embeddings. In *ASONAM*, pp. 258–265, 2017.

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 *WSDM*, pp. 459–467, 2018.

Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Chi Wang, Kuansan Wang, and Jie Tang. NetSMF: Large-scale network embedding as sparse matrix factorization. In *WWW*, pp. 1509–1520, 2019.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 2008.

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

Jengnan Tzeng, Henry Lu, and Wen-Hsiung Li. Multidimensional scaling for large genomic data sets. *BMC bioinformatics*, 9:179, 02 2008. doi: 10.1186/1471-2105-9-179.

Ulrike von Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.

Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, Jan 2015a.

Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015b.

R. Zafarani and H. Liu. Social computing data repository at ASU, 2009.

Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. Prone: Fast and scalable network representation learning. In *IJCAI-19*, pp. 4278–4284, 7 2019.

Z. Zhang, P. Cui, H. Li, X. Wang, and W. Zhu. Billion-scale network embedding with iterative random projection. In *ICDM*, pp. 787–796, 2018.