



**HAL**  
open science

# The Deep Latent Position Topic Model for Clustering and Representation of Networks with Textual Edges

Rémi Boutin, Pierre Latouche, Charles Bouveyron

► **To cite this version:**

Rémi Boutin, Pierre Latouche, Charles Bouveyron. The Deep Latent Position Topic Model for Clustering and Representation of Networks with Textual Edges. 2023. hal-04068665v3

**HAL Id: hal-04068665**

**<https://hal.science/hal-04068665v3>**

Preprint submitted on 13 Feb 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## ARTICLE TYPE

# The Deep Latent Position Topic Model for Clustering and Representation of Networks with Textual Edges

Rémi Boutin<sup>1</sup> | Pierre Latouche<sup>2</sup> | Charles Bouveyron<sup>3</sup>

<sup>1</sup>Laboratoire MAP5, CNRS, UMR 8145,  
Université Paris Cité, Paris, France

<sup>2</sup>Laboratoire LMBP, CNRS, UMR 6620,  
Université Clermont Auvergne, Aubière, France

<sup>3</sup>Laboratoire J.A.Dieudonné, CNRS, Maasai team,  
Université Côte d'Azur, INRIA, Nice, France

**Correspondence**

Corresponding author Rémi Boutin,  
Email: remi.boutin.stat@gmail.com

**Present address**

Université Paris Cité, Campus Saint-Germain,  
MAP5, 45 rue des Saints-Pères, 75006 Paris, France.

**Abstract**

Numerical interactions leading to users sharing textual content published by others are naturally represented by a network where the individuals are associated with the nodes and the exchanged texts with the edges. To understand those heterogeneous and complex data structures, clustering nodes into homogeneous groups as well as rendering a comprehensible visualisation of the data is mandatory. To address both issues, we introduce Deep-LPTM, a model-based clustering strategy relying on a variational graph auto-encoder approach as well as a probabilistic model to characterise the topics of discussion. Deep-LPTM allows to build a joint representation of the nodes and of the edges in two embeddings spaces. The parameters are inferred using a variational inference algorithm. We also introduce IC2L, a model selection criterion specifically designed to choose models with relevant clustering and visualisation properties. An extensive benchmark study on synthetic data is provided. In particular, we find that Deep-LPTM better recovers the partitions of the nodes than the state-of-the-art ETSBM and STBM. Eventually, the emails of the Enron company are analysed and visualisations of the results are presented, with meaningful highlights of the graph structure.

**KEY WORDS**

Graph convolutional network, embedded topic model, deep latent position model, unsupervised learning

## 1 | INTRODUCTION AND RELATED WORK

Numerical interactions between individuals often imply the creation of texts. For instance, on social media such as Twitter, it is possible to publish some content, a tweet or a post, that will in turn be republished, or re-tweeted, by other accounts. Also, it is possible to mention another account directly in the publication. In the same way, the exchange of mails between collaborators can be seen as connections between accounts exchanging documents. Both examples can be represented by a network with the nodes corresponding to the accounts, and the edges to the exchanged texts. Such data structure is particularly difficult to apprehend, due to the heterogeneity and the volume of the data. One solution is to cluster homogeneous nodes into groups to obtain intelligible and useful information. However, very few methods performing node clustering are actually able to simultaneously exploit both the texts present on the edges, and the connections.

In the following, we first present the advancements in the statistical network analysis field. We then review some of the core probabilistic models that can capture the main topics in a corpus of texts. Eventually, we close this section with models considering both texts and networks to cluster nodes, before introducing the contribution of the present work.

### *Statistical network analysis*

Network analysis first started with heuristic based methods. To obtain uncertainty estimates as well as more robust models, probabilistic modelling was developed in the second half of the 20th century. Statistical network analysis aims at capturing information from the network structure and making it intelligible for Human beings. One of the core model to perform model-based clustering on networks has been the stochastic block model (SBM) [1, 2, 3]. It assumes that the nodes are grouped into clusters and that the edges are independent of the nodes given their respective clusters. This model allows to discover any connectivity pattern. Another line of work has focused on obtaining relevant representations of the data in a Euclidean space. For

instance, the latent position model (LPM, [4]) obtains an informative representation of the network in a Euclidean space by assuming that each node can be modelled by a Gaussian variable in a low dimensional space. LPM was extended to the latent position clustering model (LPCM, [5]). The authors assume that the latent node positions are sampled from a mixture of normal distributions. This extension incorporates the clustering within the model and therefore combines the clustering task with the network representation. A review of standard models for statistical network analysis can be found in [6] and [7].

#### *Deep probabilistic model for graph clustering*

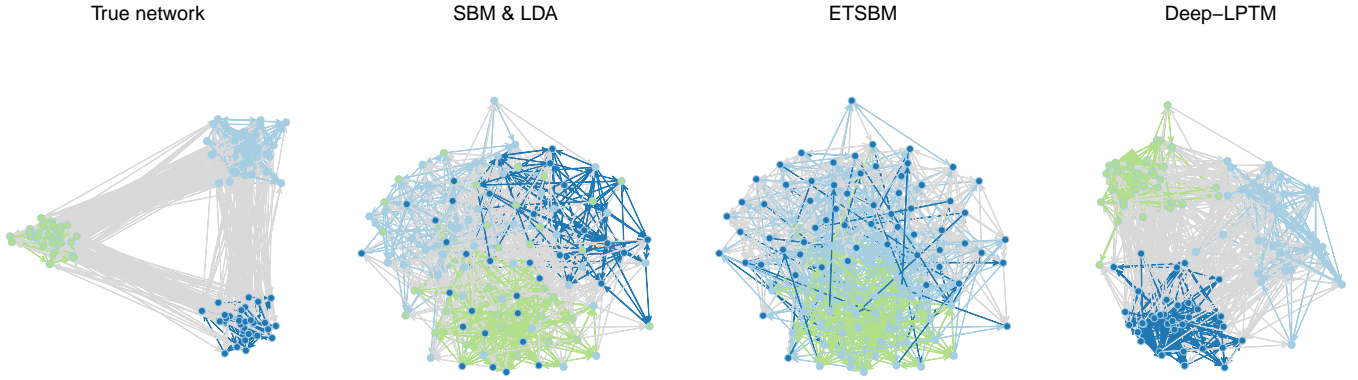
Deep neural networks (DNN) have been introduced to perform clustering in the Euclidean setting, as described in [8]. The use of DNN, as a parametric function to encode data into a latent representation, has soared thanks to the variational auto encoders (VAE). This method relies on variational Bayes inference and the reparametrisation trick [9, 10]. The interested reader can refer to [11]. Unfortunately, most traditional VAE can only be applied to Euclidean data, and, in particular, cannot be employed for network analysis. This issue was solved with the variational graph auto-encoder (VGAE, [12]) that first combined the advancement in graph neural networks, namely the graph convolutional networks (GCN, [13, 14]) and deep probabilistic models. VGAE improves upon state-of-the-art methods for the link prediction task and is capable of using node features. In [15], the authors regularise VGAE by using an adversarial inference strategy to improve the results. Unfortunately, those models rely on external methods, such as k-means, on the posterior node representations, to achieve node clustering. As an answer to this limit, [16] presented an extension of the overlapping stochastic block model [17]. They proposed to encode the node embeddings with a NN. Lately, [18] combined the two and introduced the deep latent position model (Deep-LPM). Indeed, the model relies on VGAE and assumes that the latent representations are distributed according to a mixture of Gaussians, depending on the node cluster memberships. Their results advocate the incorporation of the clustering into the latent representations.

#### *Topic Modelling*

Like network analysis, the field of topic modelling was first developed using heuristics and was only studied from a statistical perspective at the end of the 20th century. [19] provided statistical foundations for the latent semantic index (LSI) and [20] introduced a mixture model to represent the vocabulary distribution where each component modelled a specific topic. Ultimately, the latent Dirichlet allocation model (LDA) was proposed by [21]. They assumed that the topic proportions of the documents are sampled from a Dirichlet distribution. Many extensions of LDA have been proposed, including the use of a normal-logistic prior to better model the correlations between topics [22]. In the end, the gap between deep generative modelling and topic models was bridged by [23] who introduced a variational distribution parametrised by a DNN. More recently, the embedded topic model (ETM, [24]) allowed to use embeddings in order to represent both the words and the topics in the same vector space. They are used as part of the decoder and can be pre-trained on large datasets to incorporate semantic meaning, as illustrated in the original paper with examples relying on continuous bag of words [25].

#### *Joint analysis of network and texts*

While both topic modelling and statistical network analysis gave rise to many publications over the last 20 years, only few works have combined the two approaches. The community-user topic model (CUT, [26]) added a latent variable to the author-topic model (AT, [27]) to represent the communities either as a set of co-authors or as a set of topics. Thereafter, the community-author-recipient-topic model (CART, [28]) used communities both at the document generation level and at the author and recipient generation level which corresponds to the network generation. However, the high number of parameters combined with the inference based on a Gibbs sampler does not allow to scale those models to large datasets. The topic-link LDA, presented in [29], also offers a joint-analysis of texts and links in a unified framework by conditioning the generation of a link on both the topics within the documents and the community of authors. The inference relies on a variational EM algorithm to scale the approach to large datasets. However, this method only deal with undirected networks. Finally, the topic-user-community models (TUCM) was introduced in [30] and was able to discover topic-meaningful communities. The main feature of this model was its capacity to incorporate different types of interactions, well-suited for social networks applications. The inference relied on Gibbs sampling approach which can be limiting when dealing with large datasets. More recently, the stochastic topic block model (STBM) presented in [31] was the first model to handle the simultaneous clustering of nodes and edges while keeping the inference tractable to large datasets thanks to a variational classification EM based inference. This model was extended in [32] for the simultaneous clustering of bipartite networks with textual edges. It was also adapted for dynamic networks in [33]. Recently, [34] introduced the embedded topic in the stochastic block model (ETSBM) to incorporate embeddings and semantic meanings into the word representations. However, despite their modelling qualities, all these approaches do not allow for the construction of a meaningful representation of the data which characterises both the connections and the documents exchanged.



**FIGURE 1** Illustration of Deep-LPTM main contributions on a synthetic network. The colours of the nodes and of the edges denote the node clusters and the main topics of the corresponding documents respectively. The true node partition as well as the true topics of the documents are represented on the left-hand side. The three other figures are based on the respective results of SBM and LDA (second figure), ETSBM (third figure) and Deep-LPTM (fourth figure). Only Deep-LPTM is able to provide node positions incorporating information about the network structure as well as the documents content. The four figures were obtained with the *gplot* function from the *sna* library [35]. While the first network is plotted manually to highlight the generative structure, the two networks in the middle are based on the Fruchterman-Reingold algorithm while the fourth graph uses the node positions estimated by Deep-LPTM. Contrary to former methods STBM and ETSBM, which require the use of an exterior method, our methodology provides an end-to-end method tailored for representing networks with clustering properties.

### *Main contributions and organisation of the paper*

The model proposed in the present paper is the first to simultaneously cluster the nodes of a network, uncover the topics in the texts exchanged between the nodes and to output a representation of both the topics and the edges in a Euclidean space. To this aim, **(i)** we propose a generative model assuming that each node and each edge is represented in a latent space by a mixture of Gaussians. By doing so, we incorporate the clustering in the generative process such that each mixture component models either a cluster of nodes or the documents exchanged between a pair of clusters of nodes. Moreover, our model distinguishes itself from former methods by allowing each node or edge to be represented by a latent position and not only by the group it belongs to. This is illustrated in Figure 1, where the first graph depicts the true node clusters and main topics of the corresponding documents of a simulated dataset. The second graph gives, for this dataset, the node clusters provided by SBM and the topics estimated by LDA. SBM does not retrieve the true node partition and does not provide node positions to apprehend the results. An external algorithm, namely the Fruchterman-Reingold algorithm [36], considering the presence of connections only in the network, had to be used for graph representation. The third graph is based on state-of-the-art ETSBM and does not recover the node partition either. Moreover, ETSBM is not able to render a comprehensive representation of the full network. Again, as for SBM, the Fruchterman-Reingold algorithm had to be used for graph representation. Finally, the figure on the right-hand side presents the Deep-LPTM results. As opposed to the previous methods, Deep-LPTM is able to gather the information about the network structure as well as the exchanged documents into the node positions while finding the true node partition and topics. A representation of the graph is directly obtained by the estimation procedure such that no external graph representation algorithm is needed. As we shall see, the node positions are computed by considering both the connections and the content of the corresponding documents. **(ii)** We derive a two-stage variational expectation-maximisation (VEM) algorithm for estimating the model parameters as well as the posterior parameters of the latent positions. The first stage relies on analytical formulas to update the cluster probabilities as well as the mixture parameters. The second stage uses a stochastic gradient descent algorithm to update the expected lower bound with respect to the VGAE parameters and the deep topic model parameters. In particular, the deep topic model can make the best out of pre-trained embeddings. Thus, introducing semantic meaning into the word representations is possible as well as learning the representation from scratch. **(iii)** In order to choose the relevant numbers of clusters and latent space dimensions, we introduce the integrated classification and latent likelihood (IC2L) for model selection. It extends the integrated classification likelihood criterion, which was conceived for mixture models, to account for the latent representations of the nodes and of the edges. The criterion relevance is strongly upheld by the evaluation on synthetic data as well as the provided real-word use case. Moreover, by selecting a low dimensions regarding the node embedding space, IC2L praises for models with a strong and direct capacity of representation.

We close this section with the organisation of the paper. In Section 2, we present the assumptions concerning the generation of the data. The inference as well as the model selection criterion are presented in Section 3. In Section 4, Deep-LPTM and the impact of the initialisation are evaluated on synthetic data. An extensive benchmark study against state-of-the-art methods is also provided. Eventually, the emails of the Enron company are analysed with Deep-LPTM in Section 5. The results as well as the visualisations are presented to illustrate the ease of interpretation of the model outcomes.

## 1.1 | Notations

In this paper, we are interested in data represented by a graph  $\mathcal{G} := \{\mathcal{V}, \mathcal{E}\}$  where  $\mathcal{V} = \{1, \dots, N\}$  denotes the set of vertices. The set  $\mathcal{E}$  denotes the edges between the nodes with  $M = |\mathcal{E}|$  the number of edges. We focus on binary adjacency matrix  $A \in \mathcal{M}_{N \times N}(\{0, 1\})$  such that  $A_{ij}$  equals 1 if  $(i, j) \in \mathcal{E}$ , and 0 otherwise. The graph is assumed to be directed and without any self loop. Therefore  $A_{ii} = 0$  for all  $i \in \mathcal{V}$ . Finally,  $Q$  denotes the number of clusters of nodes.

Each edge in the graph represents a textual document sent from one node to another. An edge from node  $i$  to node  $j$  exists or equivalently  $(i, j) \in \mathcal{E}$ , if and only if node  $i$  sent a textual document to node  $j$ , denoted  $W_{ij}$ . We use a bag-of-words representation of the texts where  $W_{ij} = (W_{ij}^1, \dots, W_{ij}^V) \in \mathbb{N}^V$  denotes the vector of word occurrences in the document between nodes  $i$  and  $j$  such that  $W_{ij}^v$  is the number of times word  $v$  appears in the document,  $M_{ij} = \sum_{v=1}^V W_{ij}^v$  is the total number of words in document  $W_{ij}$  and  $V$  the size of the vocabulary. Hence,  $A_{ij} = 1$  if  $W_{ij}$  exists. The set of documents will be denoted  $W := (W_{ij})_{(i,j) \in \mathcal{E}}$  and the number of topics is denoted by  $K$ . Eventually, the simplex of dimension  $d$  will be denoted  $\Delta_{d-1}$ .

## 2 | MODEL

In the following, the assumptions about the graph generation as well as the hypothesis concerning the documents construction are presented.

### 2.1 | Graph generation

Assuming that the number of clusters  $Q$  is fixed before hand, each node  $i$  is assumed to belong to a cluster, represented by the cluster membership variable  $C_i$ . The variables  $C_i$ , for any  $i \in \mathcal{V}$ , are assumed to be independent and identically distributed (i.i.d) according to a multinomial distribution such that for any node  $i \in \{1, \dots, N\}$ :

$$C_i \sim \mathcal{M}_Q(1, \pi), \quad (1)$$

with  $\pi \in \Delta_{Q-1}$  and  $C_i \in \{0, 1\}^Q$  being one hot encoded so that  $C_{iq} = 1$  if node  $i$  belongs to cluster  $q$  and  $C_{iq} = 0$  otherwise. Thus, denoting  $C = (C_1, \dots, C_N)^T \in \mathcal{M}_{N \times Q}(\{0, 1\})$  the cluster membership matrix, we have:

$$p(C \mid \pi) = \prod_{i=1}^N \prod_{q=1}^Q \pi_q^{C_{iq}}. \quad (2)$$

Moreover, given its cluster membership, the node  $i$  is assumed to be represented by a Gaussian vector  $Z_i$  in a  $p$  dimensional latent space such that:

$$Z_i \mid C_{iq} = 1 \sim \mathcal{N}(\mu_q, \sigma_q^2 I_p). \quad (3)$$

Eventually, the connection between two nodes is assumed to depend on the closeness of the node representations in the latent space. Therefore, denoting  $\eta_{ij} := \kappa - \|Z_i - Z_j\|$ , which accounts for the opposite of the distance between the node latent representations, the probability for node  $i$  to be connected to node  $j$  is:

$$A_{ij} \mid Z_i, Z_j, \kappa \sim \text{Ber}\left(\frac{1}{1 + e^{-\eta_{ij}}}\right), \quad (4)$$

where a logistic function is used as a link function. For the sake of brevity, we will denote  $p_{ij} = (1 + e^{-\eta_{ij}})^{-1}$ . The probability of the entire adjacency matrix is given by:

$$p(A | Z, \kappa) = \prod_{i \neq j} p_{ij}^{A_{ij}} (1 - p_{ij})^{1 - A_{ij}}.$$

Finally, the joint-distribution of the adjacency matrix, the latent node vectors, as well as the cluster memberships can be factorised as follow:

$$p(A, Z, C | \kappa, \mu, \sigma, \pi) = p(A | Z, \kappa) p(Z | C, \mu, \sigma) p(C | \pi). \quad (5)$$

where  $\mu = (\mu_q)_q$  and  $\sigma = (\sigma_q)_q$ . It is worth noticing that the model described in Equations (1), (3) and (4) corresponds to the latent position cluster model [5]. The fundamental difference with our approach for this part of the model will arise in the inference, as discussed in Section 3.

## 2.2 | Generation of the texts on the edges

At the core of our approach is the motivation to be able to use textual data to obtain more homogeneous and meaningful clusters.

To begin with, we make the assumption that each edge can be represented in a latent space by a Gaussian vector, depending only on the node cluster memberships. Thus, given  $(C_i)_{i \in \mathcal{V}}$ , the latent variables  $Y_{ij}$  are assumed to be i.i.d such that:

$$Y_{ij} | A_{ij} C_{iq} C_{jr} = 1 \sim \mathcal{N}(m_{qr}, s_{qr}^2 I_K), \quad \forall (i, j) \in \mathcal{E}, \quad (6)$$

where  $m_{qr} \in \mathbb{R}^K$ ,  $s_{qr} \in \mathbb{R}^+$ .

Moreover, we assume that the topic proportions of the document  $W_{ij}$ , denoted  $\theta_{ij}$ , can be deduced from the latent variables such that:

$$\tilde{Y}_{ij} = \text{softmax}(Y_{ij}), \quad (7)$$

where  $\text{softmax}(x) = \left( \sum_{f=1}^F e^{x_f} \right)^{-1} (e^{x_1}, \dots, e^{x_F})^\top$  if  $x \in \mathbb{R}^F$ .

Hence, assuming that the documents are i.i.d given their corresponding topic proportions, we have for any edge  $(i, j) \in \mathcal{E}$ :

$$W_{ij} | A_{ij} = 1, \theta_{ij} \sim \mathcal{M}_V(M_{ij}, \beta^\top \theta_{ij}), \quad (8)$$

where  $\beta_k = \text{softmax}(\rho^\top \alpha_k) \in \mathbb{R}^V$ ,  $\beta = (\beta_1 \dots \beta_K)^\top \in \mathcal{M}_{K \times V}(\mathbb{R})$ ,  $\rho \in \mathcal{M}_{L \times V}(\mathbb{R})$ ,  $\alpha_k \in \mathbb{R}^L$  and  $\alpha = (\alpha_1 \dots \alpha_K) \in \mathcal{M}_{L \times K}(\mathbb{R})$ . Thus, denoting  $m = (m_{qr})_{qr}$ ,  $s = (s_{qr})_{qr}$ , the joint likelihood of the texts, the latent representation of the documents, as well as the clusters memberships can be computed as:

$$p(W, Y | A, C, \rho, \alpha, m, s) = p(W | A, Y, \rho, \alpha) p(Y | A, C, m, s), \quad (9)$$

where  $m = (m_{qr})_{1 \leq q, r \leq Q}$  and  $s = (s_{qr})_{1 \leq q, r \leq Q}$ .

Figure 2 gives a graphical representation of the generative assumptions presented above. We omitted the parameters for the sake of clarity but the full version can be found in Appendix A1. Interestingly enough, those assumptions can be linked to existing models as we shall see in the next section.

## 2.3 | Link with other models

On the one hand, if the topic modelling alone is considered, restricting all topic proportions to be equal for all  $(i, j)$  such that  $C_{iq} C_{jr} = 1$  corresponds to the text modelling in ETSBM [34]. In that sense, Deep-LPTM increases the freedom of each edge representation compared to ETSBM. Additionally, Deep-LPM corresponds to Deep-LPTM when the textual data present on the edges are disposed of (or LPCM if no GCN-based encoder is used in the inference strategy). Accordingly, Deep-LPTM prolongs Deep-LPM and LPCM to networks with textual data. On the other hand, discarding the information provided by the graph and the clustering of the nodes would correspond to ETM applied on the observed documents. Therefore, Deep-LPTM extends ETM to texts with a connectivity structure to improve the topic modelling.

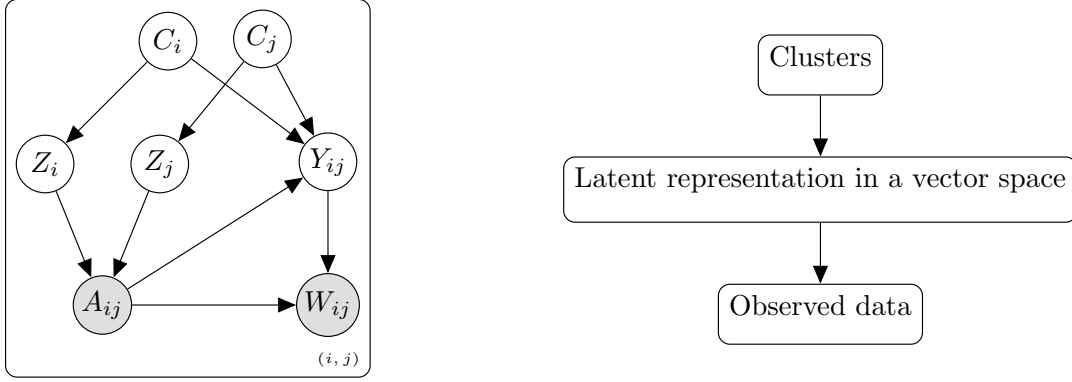


FIGURE 2 Graphical representation of the model without the parameters for the sake of clarity.

### 3 | INFERENCE

In the next section, the inference of the model is presented as well as the model selection criterion.

#### 3.1 | Likelihood

In this work, we consider the marginal likelihood of the network and the texts for parameter estimation. The latent variables are denoted by  $C = (C_i)_{i=1}^N$ ,  $Z = (Z_i)_{i=1}^N$  and  $Y = (Y_{ij})_{(i,j) \in \mathcal{E}}$ , and the set of parameters is  $\Theta = \{\pi, \mu, \sigma, \kappa, m, s, \alpha, \rho\}$ . Thus, the marginal log-likelihood is given by:

$$\mathcal{L}(\Theta; A, W) = \log p(A, W | \Theta) = \log \left( \sum_C \int_Z \int_Y p(A, W, C, Z, Y | \Theta) dZ dY \right). \quad (10)$$

Unfortunately, this quantity is not tractable since the sum over  $C$  requires to compute  $Q^N$  terms. Besides, it involves integrals that cannot be computed analytically. Therefore, we choose to rely on a variational inference approach for approximation purposes.

##### 3.1.0.1 | Decomposition of the marginal log-likelihood

For any distribution  $R(C, Z, Y)$ , the following decomposition holds:

$$\mathcal{L}(\Theta; A, W) = \mathcal{L}(R(\cdot); \Theta) + \text{KL}(R(\cdot) \| p(C, Z, Y | A, W)), \quad (11)$$

where

$$\mathcal{L}(R(\cdot); \Theta) = \mathbb{E}_R \left[ \log \frac{p(A, W, C, Z, Y | \Theta)}{R(C, Z, Y)} \right]. \quad (12)$$

Since the Kullback-Leibler divergence is always positive in Equation (11), the ELBO  $\mathcal{L}(R(\cdot); \Theta)$  is a lower bound of the marginal log-likelihood. Moreover, the closer  $R(\cdot)$  is to the posterior distribution of the latent variables, in terms of Kullback-Leibler divergence, the closer the ELBO is to the marginal log-likelihood. Since the marginal log-likelihood does not depend on  $R(\cdot)$ , maximizing the ELBO with respect to  $R(\cdot)$  is equivalent to minimizing the Kullback-Leibler divergence between  $R(\cdot)$  and the posterior distribution. To make the ELBO tractable, we restrict the family of variational distributions by considering a mean field assumption. Moreover, the assumptions in Equations (14) to (16) allow us to take full advantage of the SGD efficiency to optimise the variational distribution while keeping a high flexibility thanks to the neural network parametrisations of the

distributions (see, section 2.5 and 2.6 of [37]). The assumptions are given by:

$$R(C, Z, Y | A, W) = R(C)R(Z | A)R(Y | A, W), \quad (13)$$

$$R(C) = \prod_{i=1}^N R_{\tau_i}(C_i) = \prod_{i=1}^N \mathcal{M}_Q(C_i; 1, \tau_i), \quad (14)$$

$$R(Z | A) = \prod_{i=1}^N R_{\phi_Z}(Z_i | A) = \prod_{i=1}^N \mathcal{N}(Z_i; \mu_{\phi_Z}(A)_i, \sigma_{\phi_Z}^2(A)_i I_P), \quad (15)$$

$$R(Y | A, W) = \prod_{i \neq j} R_{\phi_Y}(Y_{ij} | W_{ij})^{A_{ij}} = \prod_{i \neq j} \mathcal{N}(Y_{ij}; \mu_{\phi_Y}(W_{ij}), \text{diag}(\sigma_{\phi_Y}^2(W_{ij})))^{A_{ij}}, \quad (16)$$

where  $\tau = (\tau_i)_{i=1}^N$  with  $\forall i \in \{1, \dots, N\}$ ,  $\tau_i \in \Delta_{Q-1}$ . Moreover, in Equation 15, the mapping  $\mu_{\phi_Z} : \mathcal{M}_{N \times N}(\mathbb{R}) \mapsto \mathcal{M}_{N \times P}(\mathbb{R})$  ( $\sigma_{\phi_Z}^2 : \mathcal{M}_{N \times N}(\mathbb{R}) \mapsto (\mathbb{R}^+)^N$  respectively) is the mapping normalising the adjacency matrix by its degree,  $D^{-1/2}AD^{-1/2}$ , and encoding the normalised adjacency matrix into the approximated posterior means (standard deviations) of the node latent positions. The diagonal matrix  $D$  is filled with  $D_{ii}$ , the degree of node  $i$ , for all nodes. The two mappings  $\mu_{\phi_Z}$  and  $\sigma_{\phi_Z}^2$  rely on a GCN parametrised by  $\phi_Z$  [12]. Similarly, in Equation 16,  $\mu_{\phi_Y} : \mathcal{M}_{M \times V}(\mathbb{R}) \mapsto \mathcal{M}_{M \times K}(\mathbb{R})$  ( $\sigma_{\phi_Y}^2 : \mathcal{M}_{M \times V}(\mathbb{R}) \mapsto \mathcal{M}_{M \times K}(\mathbb{R}^+)$  respectively) encodes the documents into the approximated posterior means (standard deviations) of their corresponding latent vectors. However, the two functions rely on the ETM encoder with parameter  $\phi_Y$  [24]. In practice, in all the experiments we carried out, we used a feed-forward neural network to encode the documents, with three layers and 800 units on each layer. The first two layers are shared to encode the variances and the means while the last layer is specific to each vector. Regarding the encoder of the adjacency matrix, we rely on [12] such that,  $\mu_{\phi_Z}(A) = \tilde{A} \text{ReLU}(\tilde{A}W_0)W_\mu$  and  $\log \sigma_{\phi_Z}^2(A) = \tilde{A} \text{ReLU}(\tilde{A}W_0)W_\sigma$ , where  $\tilde{A} = D^{-1/2}AD^{-1/2}$  and  $\text{ReLU}(x) = (\max(0, x_1), \dots, \max(0, x_F))$  if  $x \in \mathbb{R}^F$ .  $\mu_{\phi_Z}(\cdot)$  and  $\log \sigma_{\phi_Z}^2(\cdot)$  share the first-layer parameter  $W_0 \in \mathcal{M}_{N \times D}$  with  $D = 10$  in all the experiments we carried out, and  $W_\mu, W_\sigma \in \mathcal{M}_{D \times P}$ . For the sake of brevity, we will take the exponential of the encoder of the log variance and consider  $\sigma_{\phi_Z}^2(\cdot)$ .

Thus, the ELBO can be decomposed as follow :

$$\begin{aligned} \mathcal{L}(R(\cdot); \Theta) &= \mathbb{E}_R [\log p(A | Z, \kappa)] + \mathbb{E}_R [\log p(W | A, Y, \rho, \alpha)] + \mathbb{E}_R [\log p(C | \pi)] \\ &\quad + \mathbb{E}_R [\log p(Z | C, \mu, \sigma)] + \mathbb{E}_R [\log p(Y | A, C, m, s)] - \mathbb{E}_R [\log R(C)] \\ &\quad - \mathbb{E}_R [\log R(Z | A)] - \mathbb{E}_R [\log R(Y | A, W)]. \end{aligned} \quad (17)$$

The computation of each term in Equation (17) is detailed in Appendix B. To optimise the ELBO, we propose here to alternate between closed form updates and stochastic gradient descent steps thanks to the results presented in the next section.

## 3.2 | Optimisation

### 3.2.1 | Analytical updates

Given a variational distribution  $R(\cdot)$  complying with Equations (13) to (16), the model parameters can be updated using Propositions (1) and (2).

**Proposition 1.** *Let  $R(\cdot)$  be a variational distribution complying with Equations (13) to (16). The parameters of the nodes embeddings distributions maximising the ELBO are given by:*

$$\mu_q = \frac{1}{N_q} \sum_{i=1}^N \tau_{iq} \mu_{\phi_Z}(A)_i, \quad (18)$$

$$\sigma_q^2 = \frac{1}{pN_q} \sum_{i=1}^N \tau_{iq} (p\sigma_{\phi_Z}^2(A)_i + \|\mu_{\phi_Z}(A)_i - \mu_q\|_2^2), \quad (19)$$

where  $N_q = \sum_{i=1}^N \tau_{iq}$  is the posterior mean of the number of nodes in cluster  $q$ .

The proof is given in Appendix C.1.0.3. Interestingly, this proposition states that the  $\mu_q$  are the weighted mean of the (approximated) posterior mean nodes positions  $\mu_{\phi_Z}$  provided by the DNN. It also indicates that the  $\sigma_q$  are updated as the sum



of two terms: the first one corresponds to a weighted mean of the posterior variances while the second one is the intra cluster variance weighted by the posterior clusters membership probabilities  $\tau_i$ .

**Proposition 2.** *For a given variational distribution  $R(\cdot)$  complying with Equations (13) to (16), with parameters  $\tau, \mu_{\phi_Y}, \sigma_{\phi_Y}$ , the parameters of the edge embeddings distributions maximising the ELBO are given by:*

$$m_{qr} = \frac{1}{N_{qr}} \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr} \mu_{\phi_Y}(W_{ij}), \quad (20)$$

$$s_{qr}^2 = \frac{1}{KN_{qr}} \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr} \left[ \sum_{k=1}^K \sigma_{\phi_Y}^2(W_{ij})_k + \|\mu_{\phi_Y}(W_{ij}) - m_{qr}\|_2^2 \right], \quad (21)$$

where  $N_{qr} = \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr}$  denotes the expected number of documents sent from cluster  $q$  to cluster  $r$  under the approximated posterior distribution.

The proof is given in Appendix C.1.0.4 The interpretation following Proposition 1 can also be applied to Proposition 2 while a second DNN is used. The main difference lies in the weighting that here corresponds to the probability that the pair of nodes composing each edge belong to a pair of cluster. For instance, for the edge  $(i, j)$ , the posterior probability that node  $i$  belongs to cluster  $q$  and node  $j$  to cluster  $r$  is given by  $\tau_{iq} \tau_{jr}$ .

### 3.2.2 | The stochastic gradient descent

The other model parameters  $\kappa, \rho$  and  $\alpha$ , and variational parameters  $\phi_Z$  and  $\phi_Y$  cannot be updated with analytical formulas because of the integral involving the variational distribution  $R(\cdot)$  in the ELBO. In the following section, we aim at deriving estimates of the gradients of the ELBO with respect to these parameters, to perform stochastic gradient descent.

#### 3.2.2.1 | Model parameters

The partial derivatives of the ELBO with respect to each parameter  $\kappa, \rho$  and  $\alpha$  are obtained thanks to Monte-Carlo estimates. For instance, the gradient of the ELBO with respect to  $\kappa$  is estimated by:

$$\begin{aligned} \frac{\partial}{\partial \kappa} \mathcal{L}(R(\cdot); \Theta) &= \frac{\partial}{\partial \kappa} \mathbb{E}_R [\log p(A | Z, \kappa)] \\ &= \mathbb{E}_R \left[ \frac{\partial}{\partial \kappa} \log p(A | Z, \kappa) \right] \\ &\approx \frac{1}{S} \sum_{s=1}^S \frac{\partial}{\partial \kappa} \log p(A | Z^{(s)}, \kappa), \end{aligned}$$

where  $Z^{(s)} = (Z_1^{(s)}, \dots, Z_N^{(s)})$  and  $Z_i^{(s)} \stackrel{i.i.d}{\sim} \mathcal{N}(\mu_{\phi_Z}(A)_i, \sigma_{\phi_Z}^2(A)_i I_p)$ . The same computations result in estimates for the partial derivatives of the ELBO with respect to  $\rho$  and  $\alpha$ . In practice, we rely on the common practice in the field of VAE and set  $S = 1$ .

#### 3.2.2.2 | Variational parameters

The last parameters to update are the variational parameters  $\phi_Y$  and  $\phi_Z$ . Ideally, we would like to use the same computation as in the previous section. For instance, we would like to compute the following partial derivatives of the ELBO with respect to  $\phi_Z$ :

$$\begin{aligned} \frac{\partial}{\partial \phi_Z} \mathcal{L}(R(\cdot); \Theta) &= \frac{\partial}{\partial \phi_Z} \mathbb{E}_R [\log p(A | Z, \kappa) + \log p(Z | C, \mu, \sigma) - \log R(Z)] \\ &= \frac{\partial}{\partial \phi_Z} \mathbb{E}_R [\log p(A | Z, \kappa)] \end{aligned} \quad (22)$$

$$- \sum_{i=1}^N \sum_{q=1}^Q \tau_{iq} \frac{\partial}{\partial \phi_Z} \text{KL}_{iq}^Z(\mu_{\phi_Z}(A)_i, \sigma_{\phi_Z}(A)_i, \mu_q, \sigma_q), \quad (23)$$

with

$$\begin{aligned} \text{KL}_{iq}^Z(\mu_{\phi_Z}(A)_i, \sigma_{\phi_Z}(A)_i, \mu_q, \sigma_q) &= \log \frac{\sigma_q^p}{\sigma_{\phi_Z}(A)_i^p} - \frac{p}{2} \\ &\quad + \frac{p\sigma_{\phi_Z}^2(A)_i + \|\mu_{\phi_Z}(A)_i - \mu_q\|_2^2}{2\sigma_q^2}. \end{aligned}$$

See Appendix B1 for the computational details. Unfortunately, the expectation of the term on the left-hand side of Equation (22), is taken with respect to the variational distribution which depends on  $\phi_Z$ . Therefore, the derivation of this quantity is not straightforward. Thanks to [9] and [10], this difficulty can be tackled by using the reparametrisation trick. In particular, let  $\varepsilon_i$  be a centred, normalised and  $P$ -dimensional Gaussian vector. Hence, the vectors  $Z_i$  and  $\mu_{\phi_Z}(A)_i \oplus [\sigma_{\phi_Z}(A)_i \odot \varepsilon_i]$  have the same distribution. Therefore, the expectation can be taken with respect to  $\epsilon = (\varepsilon_i)_{i=1, \dots, N}$  which gives:

$$\frac{\partial}{\partial \phi_Z} \mathbb{E}_R [\log p(A | Z, \kappa)] = \frac{\partial}{\partial \phi_Z} \mathbb{E}_\epsilon [\log p(A | \mu_{\phi_Z}(A) \oplus [\sigma_{\phi_Z}(A) \odot \epsilon], \kappa)]$$

where  $\mu_{\phi_Z}(A) \oplus [\sigma_{\phi_Z}(A) \odot \epsilon] = \left( \mu_{\phi_Z}(A)_i \oplus [\sigma_{\phi_Z}(A)_i \odot \varepsilon_i] \right)_{i=1, \dots, N}$ ,  $\odot$  denotes the Hadamard product and  $\oplus$  the element-wise sum. A Monte-Carlo estimate of this quantity is derived by sampling  $S$  centred and reduced  $P$ -dimensional Gaussian vectors  $\epsilon_i^{(s)}$  with  $s = 1, \dots, S$ ,  $i = 1, \dots, N$  and with  $\epsilon^{(s)} = (\epsilon_i^{(s)})_{i=1, \dots, N}$ . Plugging it back into (22) gives the following estimate:

$$\begin{aligned} \frac{\partial}{\partial \phi_Z} \mathcal{L}(R(\cdot); \Theta) &\approx \frac{1}{S} \sum_{s=1}^S \left[ \frac{\partial}{\partial \phi_Z} \log p(A | \mu_{\phi_Z}(A) \oplus [\sigma_{\phi_Z}(A) \odot \epsilon^{(s)}], \kappa) \right] \\ &\quad - \sum_{i=1}^N \sum_{q=1}^Q \tau_{iq} \frac{\partial}{\partial \phi_Z} \text{KL}_{iq}^Z(\mu_{\phi_Z}(A)_i, \sigma_{\phi_Z}^2(A)_i, \mu_q, \sigma_q^2). \end{aligned}$$

The same derivation steps lead to a similar estimate for the partial derivatives of the ELBO with respect to  $\phi_Y$ . Thanks to the low variances of the gradients estimated with the reparametrisation trick and to avoid increasing the computations, we use a sample size  $S = 1$ , as advised in the VAE literature. In addition, the computation of the partial derivatives is implemented with Pytorch automatic differentiation framework [38] to take fully advantage of the computational efficiency of GPUs. Moreover, we rely on the Adam optimiser [39] to carry out the stochastic gradient descent with a learning rate of 0.002 (0.005 respectively) for the optimiser of  $\kappa$  and  $\phi_Z$  ( $\phi_Y$  respectively).

### 3.3 | Model selection

To complete the inference, we present IC2L, a new model selection criterion accounting for both the clustering partition as well as the latent representations. In all previous sections, we considered the number of clusters  $Q$ , the number of topics  $K$  and the dimension of the node latent space  $P$  fixed before hand. In this section, we aim at selecting the triplet  $(K, P, Q)$  that captures the most information out of the data without over-parametrisation. The integrated complete (or classification) likelihood [40], denoted ICL, was introduced for mixture models and is now a common model selection criterion in this context.

First, considering a mixture model  $\mathcal{M}$ , with observed data  $X$ , latent cluster memberships  $C$ ,  $Q$  clusters and distribution parameters in the set  $\Theta$ , the complete likelihood refers to  $p(X, C | \mathcal{M}, Q, \Theta)$ . This quantity depends on both the clustering and the model parameters. Then, to account for the uncertainty over the set of parameters and to penalise the model complexity, [40] proposed to integrate over  $\Theta$  and to evaluate the quantity  $\log p(X, C | \mathcal{M}, Q) = \int_{\theta \in \Theta} \log p(X, C | \mathcal{M}, Q, \theta) d\theta$ . Since the integral is not tractable for many statistical models, the authors relied on a BIC-like approximation of this quantity. In the present paper, we propose to extend ICL to include the evaluation of the node embeddings  $Z$ , as well as the edge embeddings  $Y$ . Denoting  $\mathcal{M}$  the model presented in Sections 2.1 and 2.2, we are interested in:

$$\log p(A, W, Z, Y, C | \mathcal{M}, Q, K, P) = \log \int_{\theta} p(A, W, Z, Y, C | \theta, \mathcal{M}, Q, K, P) p(\theta) d\theta. \quad (24)$$

Since this quantity is not tractable, we derive an estimate in the following proposition.

**Proposition 3.** *Let us consider a model  $\mathcal{M}$ , as described in Section 2, with  $Q$  denoting the number of clusters,  $K$  the number of topics and  $P$  the dimension of the node latent space. In addition, let us assume that the prior of the model parameters fully factorises, as  $p(\kappa, \pi, \mu, \sigma, m, s, \rho, \alpha) = p(\kappa)p(\pi)p(\mu)p(\sigma)p(m)p(s)p(\rho)p(\alpha)$ . Then, the IC2L( $\mathcal{M}, Q, K, P, \hat{Z}, \hat{Y}, \hat{C}$ ) criterion, denoted IC2L, is given by:*

$$\begin{aligned}
IC2L &= \max_{\theta} \log p(A, W, \hat{Z}, \hat{Y}, \hat{C} \mid \theta, \mathcal{M}, Q, K, P) - \Omega(\mathcal{M}, Q, K, P) \\
&= \max_{\kappa} \log p(A \mid \hat{Z}, \kappa, \mathcal{M}) - \frac{1}{2} \log(N(N-1)) \\
&\quad + \max_{\mu, \sigma} \log p(\hat{Z} \mid \hat{C}, \mu, \sigma, \mathcal{M}, Q, P) - \frac{QP + Q}{2} \log(N) \\
&\quad + \max_{\rho, \alpha} \log p(W \mid A, \hat{Y}, \rho, \alpha, \mathcal{M}) - \frac{VL + KL}{2} \log(M) \\
&\quad + \max_{m, s} \log p(\hat{Y} \mid A, \hat{C}, m, s, \mathcal{M}, K) - \frac{Q^2K + Q^2}{2} \log(M) \\
&\quad + \max_{\pi} \log p(\hat{C} \mid \pi, \mathcal{M}, Q) - \frac{Q-1}{2} \log(N), \tag{25}
\end{aligned}$$

with  $\hat{Z}, \hat{Y}$  and  $\hat{C}$  the maximum-a-posteriori estimates, and

$$\begin{aligned}
\Omega(\mathcal{M}, Q, K, P) &= \frac{1}{2} \log(N(N-1)) \\
&\quad + \frac{Q(P+2) - 1}{2} \log(N) \\
&\quad + \frac{L(V+K) + Q^2(K+1)}{2} \log(M), \tag{26}
\end{aligned}$$

where each term in Equation (26) corresponds to the penalisation of the BIC-like approximation of the probabilities, as detailed in Equation (25).

*Proof.* See Appendix C.2. □

Ultimately, the relevance of this criterion as well as the parameters estimation are assessed in the next section on synthetic data. Moreover, an extensive comparison with baseline methods is provided.

## 4 | NUMERICAL EXPERIMENTS

This section is dedicated to the assessment of the proposed methodology. We start with an introductory example to illustrate the results obtained with Deep-LPTM. We continue with an evaluation of the initialisation impact on our method. Then, we move on to Section 4.4 to provide numerical evidence of the robustness of IC2L against the dimensions of the parameter spaces. We close this section with a benchmark study to compare Deep-LPTM with the state-of-the-art ETSBM and STBM.<sup>‡</sup>

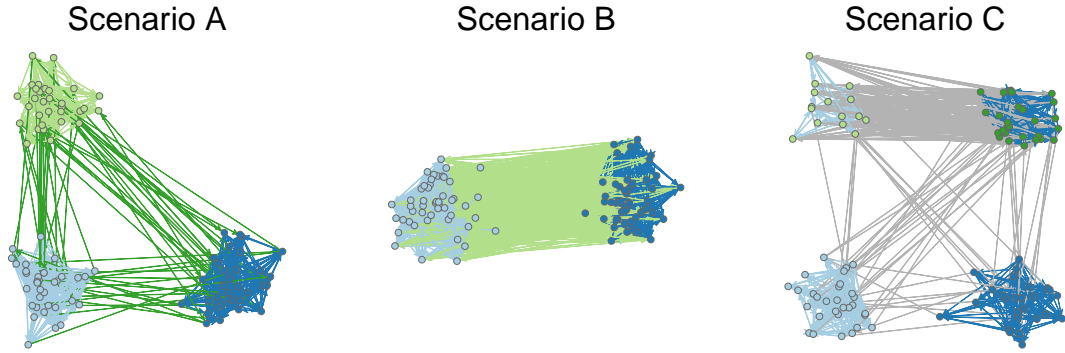
### 4.1 | Simulation settings

To begin with, we introduce three simulation scenarios to be used for evaluating the methodology on different conditions detailed hereinafter.

#### 4.1.0.1 | Scenarios

- Scenario A is constituted of three communities, each defining a cluster, and four topics. By definition, a community is a group of nodes more densely connected together than with the rest of the network. For each cluster, a specific topic is employed to sample the documents associated with the intra-cluster connections. Besides, an extra topic is employed to model documents

<sup>‡</sup> Our code is available at [https://plmlab.math.cnrs.fr/rboutin/deeplptm\\_package](https://plmlab.math.cnrs.fr/rboutin/deeplptm_package).



**FIGURE 3** Networks sampled from each scenario. The node colours denote the node cluster memberships and the edge colours denote the majority topic in the corresponding documents.

sent between nodes from different clusters. Hence, by construction, the clustering structure can be retrieved either using the network or the texts only.

- Scenario *B* is made of a single community and three topics. Consequently, all nodes connect with the same probability. Then, the nodes are spread into two clusters using distinct topics. An extra topic is used to model documents exchanged between the two clusters. Accordingly, the network itself is not sufficient to find the two clusters but the documents are.
- Scenario *C* comprises three communities and three topics. Two of the communities are associated with their respective topics, say  $t_1$  and  $t_2$ . Furthermore, following Scenario *B*, the third community is split in two clusters, one being associated with topic  $t_1$  and the other with  $t_2$ . Thus, considering both texts and topology, each network is actually made up of four node clusters. Consequently, both textual data and the network are necessary to uncover the clusters. This scenario will be of major interest in this experiment section since it ensures that the two sources of information are correctly used to uncover the node partition.

For all scenarios, networks with 100 nodes are sampled and the edges holding the documents are constructed by sampling words from four BBC articles, focusing each on a given topic. The first topic deals with the UK monarchy, the second with cancer treatments, the third with the political landscape in the UK and the last topic deals with astronomy. In the general setting, for all scenarios, the average text length for the documents is set to 150 words. The parameters used to sample data from the three scenarios are given in Table 1. Moreover, three examples of networks generated from *A*, *B* and *C* are presented in Figure 3. To summarise, the three proposed scenarios inspect different facets of the model. Scenario *A* insures that the model rightfully uses the network structure, Scenario *B* focuses on the usage of the topics to recover the node partition. Finally, Scenario *C* combines the two scenarios to guarantee that both sources of information are correctly utilised simultaneously.

	Scenario <i>A</i>	Scenario <i>B</i>	Scenario <i>C</i>
$Q$ (clusters)	3	2	4
$K$ (topics)	4	3	3
Communities	3	1	3
$\pi_{qr}$ (connection probabilities) $\psi = 0.25, \epsilon = 0.01$	$\begin{pmatrix} \psi & \epsilon & \epsilon \\ \epsilon & \psi & \epsilon \\ \epsilon & \epsilon & \psi \end{pmatrix}$	$\begin{pmatrix} \psi & \psi \\ \psi & \psi \end{pmatrix}$	$\begin{pmatrix} \psi & \epsilon & \epsilon & \epsilon \\ \epsilon & \psi & \epsilon & \epsilon \\ \epsilon & \epsilon & \psi & \psi \\ \epsilon & \epsilon & \psi & \psi \end{pmatrix}$
Topics matrix $\mathbf{T}$ between pairs of clusters $(q, r)$	$\begin{pmatrix} t_1 & t_4 & t_4 \\ t_4 & t_2 & t_4 \\ t_4 & t_4 & t_3 \end{pmatrix}$	$\begin{pmatrix} t_1 & t_3 \\ t_3 & t_2 \end{pmatrix}$	$\begin{pmatrix} t_1 & t_3 & t_3 & t_3 \\ t_3 & t_2 & t_3 & t_3 \\ t_3 & t_3 & t_1 & t_3 \\ t_3 & t_3 & t_3 & t_2 \end{pmatrix}$

**TABLE 1** Detail of the three simulation scenarios used to evaluate our model.

#### 4.1.0.2 | Clustering performance evaluation

The adjusted rand index (ARI) is used as a measure of the closeness between two partitions. In this paper, ARI compares the true node labels with the node partition provided by a model. In particular, obtaining an ARI of 0 suggests that the clustering is as

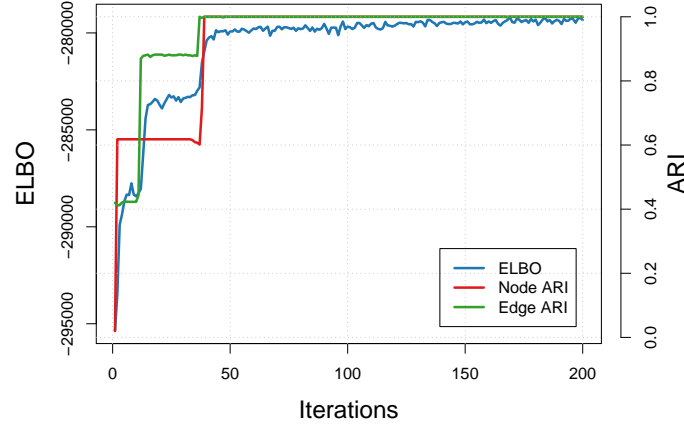


FIGURE 4 Evolution of the ELBO, as well as the node and edge ARI during the optimisation of Deep-LPTM.

close to the true node labels as a random cluster assignment of the nodes. On the contrary, the closer the ARI is to 1, the better the results are. Ultimately, an ARI of 1 signifies that the true partition was perfectly recovered (up to a label permutation).

#### 4.1.0.3 | Level of difficulty

In order to generate more situations from the three scenarios, we introduce the *Hard* difficulty to test the model robustness against two aspects. First, we want to test the model against documents using several topics. Thus, in the *Hard* difficulty, the documents are formed of multiple topics such that, for any edge  $(i, j)$  with node  $i$  in cluster  $q$  and node  $j$  in cluster  $r$ , the topic proportions are computed as a ratio between the pure topic proportions  $\tilde{Y}_{qr}^* \in \{0, 1\}^K$ , with zeros everywhere except at the coordinate corresponding to the true topic, and between the uniform distribution over the topics. This combination is controlled by a parameter  $\zeta$  such that  $\zeta = 0$  corresponds to a pure topic case while  $\zeta = 1$  leads to a uniform distribution over the topics. This translates into:

$$\tilde{Y}_{qr} = (1 - \zeta)\tilde{Y}_{qr}^* + \zeta * \left(\frac{1}{K}, \dots, \frac{1}{K}\right)^\top, \quad (27)$$

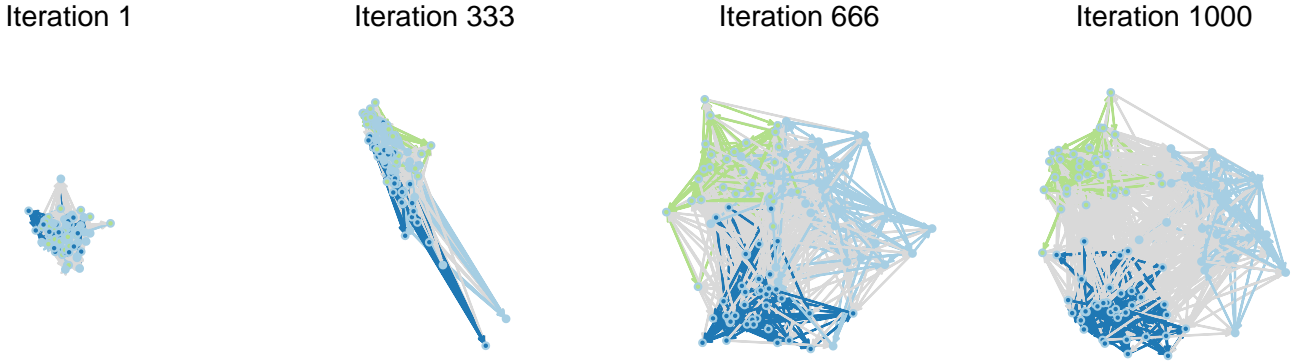
with  $\zeta = 0.7$  in the *Hard* setting. The second aspect tested by the *Hard* setting is the robustness in the presence of less connected communities. Consequently, the intra-cluster connection probability is decreased from  $\psi = 0.25$  in the classical setting to  $\psi = 0.1$  in the *Hard* one.

## 4.2 | Main features of Deep-LPTM

This section gives an overview of the main features of Deep-LPTM on one network simulated according to Scenario A, with an intra cluster connection probability  $\psi$  equal to 0.15, an inter-cluster connection probability  $\epsilon$  fixed to 0.05 and the parameter controlling the topic proportions  $\zeta$  set to 0.5. In addition,  $\phi_Y, \rho, \alpha$ , the parameters referring to the topic modelling, are pre-trained for only 5 epochs with ETM alone. Conversely, the parameters  $\phi_Z, \kappa, \mu, \sigma$ , related to network modelling, are randomly initialised without pre-training to illustrate the evolution of the node embeddings during the optimisation. In the rest of the paper, those parameters will be pre-trained by running ETM and Deep-LPM independently before hand.

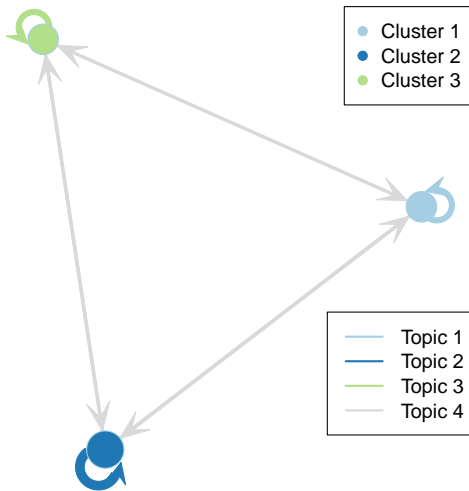
On the one hand, the evolution of the ELBO as well as the ARI of the nodes and the edges are presented in Figure 4. The node ARI and the edge ARI increase to reach an ARI of 1 following the evolution of the ELBO. We only display the first 200 epochs for the sake of clarity, but the entire training is provided in the appendix.

On the other hand, Figure 5 features the evolution of the node latent positions during the training. Interestingly enough, Deep-LPTM finds a meaningful representation of the network even with a random initialisation. This difficult problem requires to train the model longer when no initialisation is provided. Hence, the ELBO continues to increase as displayed in Figure D2, presented in the appendix. In the rest of the paper, the GCN parameters as well as the topic model parameters are initialised



**FIGURE 5** Evolution of the latent node positions during the training of Deep-LPTM.

before hand. In addition, the node cluster memberships probabilities are initialised with a similarity-based method between the topic proportions  $\tilde{Y}$  of the node neighbours [31].



**FIGURE 6** The meta-network is a representation of the network at the cluster level based on Deep-LPTM estimates. Each cluster is represented by a node, the node sizes depend on the number of nodes assigned to each cluster, the node positions as well as the major topics between two connected clusters (denoted by the colours of the edges) are estimated by the parameters  $(\mu_q)_q$  as well as  $(m_{qr})_{qr}$  respectively and the sizes of the edges depend on the number of connections between two clusters.

	1	2	3	4
1	cancer	black	princess	seats
2	cell	hole	birth	david
3	occur	gravity	charlotte	political
4	genes	light	cambridge	lost
5	cancers	shadow	queen	kingdom
6	due	credit	granddaughter	black
7	mutations	event	duchess	party
8	radiation	disc	palace	part
9	princess	princess	london	resentment
10	include	horizon	great	united

**TABLE 2** Topics of the model in Scenario A Easy, represented by the 10 most probable words per topic.

The meta-network, represented in Figure 6, describes the connectivity at the cluster level. The node sizes depend on the number of nodes assigned to each cluster  $q$  and are given by  $\sum_{i=1}^N \hat{C}_{iq}$ , with  $\hat{C}_{iq} = 1$  if  $\text{argmax } \tau_i = q$  and 0 otherwise. The cluster positions are estimated by  $(\mu_q)_q$  and the major topics (denoted by the colours of the edges on the figure) between two clusters is estimated by the  $\text{argmax } m_{qr}$  for all pairs of clusters  $(q, r)$ . Finally, the sizes of the edges depend on the number of connections between clusters, given by  $\sum_{i,j=1}^N \hat{C}_{iq} \hat{C}_{jr}$  for all pairs of clusters  $(q, r)$ .

		ScenarioA	ScenarioB	ScenarioC
		Node ARI	Node ARI	Node ARI
Easy	Random init	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	Dissimilarity init	0.97 ± 0.06	1.00 ± 0.00	0.98 ± 0.03
	Deep-LPTM random	1.00 ± 0.00	1.00 ± 0.01	0.63 ± 0.20
	Deep-LPTM dissim	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Deep-LPTM - PT random	1.00 ± 0.00	0.90 ± 0.30	0.55 ± 0.15
	Deep-LPTM - PT dissim	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Hard	Random init	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	Dissimilarity init	0.31 ± 0.14	1.00 ± 0.00	0.38 ± 0.24
	Deep-LPTM random	0.80 ± 0.21	0.95 ± 0.05	0.47 ± 0.02
	Deep-LPTM dissim	0.99 ± 0.02	1.00 ± 0.00	0.89 ± 0.15
	Deep-LPTM - PT random	0.95 ± 0.05	0.73 ± 0.30	0.45 ± 0.04
	Deep-LPTM - PT dissim	1.00 ± 0.01	1.00 ± 0.00	0.85 ± 0.18

**TABLE 3** Adjusted rand index (ARI) of the initialisations and the results of Deep-LPTM in terms of node clustering, without and with pre-trained embeddings (denoted PT in that case). ARI is averaged over 10 graphs, for each scenario and difficulty.

Eventually, Table 2 presents the topics obtained by Deep-LPTM. They are both very interpretable and distinguishable one from another which is crucial to understand complicated datasets. This will be stressed in the analysis of Enron email dataset in Section 5.

### 4.3 | Impact of the initialisation and the pre-trained embeddings

This section aims at evaluating the improvement of our method upon the initialisation, with a warm start and without. In this regard, Table 3 presents the ARI of a random initialisation as well as a dissimilarity initialisation [31], denoted *random* and *dissimilarity* respectively, in Table 3. The initialisation alone (without any model name preceding it in the table) as well as the model with the initialisations (with the model preceding the initialisation) are provided. Moreover, the Deep-LPTM node clustering is evaluated with and without pre-trained skipgram embeddings [25], denoted *PT* in the table. The results are obtained by averaging the ARI over 10 graphs for each scenario and difficulty and can be summarised in three points.

First, in all cases where the initialisation has not already reached an ARI of 1, Deep-LPTM improves the node clustering, even in difficult settings with no warm-start. For instance, in Scenario *A* with the *Hard* setting, the model starts from an ARI of 0.31, with the dissimilarity initialisation, to reach 0.99 and 1.00 without and with pre-trained embeddings respectively.

Second, the improvement provided by the pre-trained embeddings depends on the scenario. On the one hand, Scenario *A* benefits from the usage of pre-trained embeddings which always improves the results. For instance, in the *Hard* setting with a random initialisation, the ARI increases from  $0.80 \pm 0.21$  to  $0.95 \pm 0.05$ . On the other hand, Scenario *B* and *C* always favour the results without pre-trained embeddings. As an example, with a random initialisation in the *Hard* setting, the ARI decreases from  $0.95 \pm 0.05$  to  $0.73 \pm 0.30$  and  $0.47 \pm 0.02$  to  $0.45 \pm 0.04$  in Scenario *B* and *C* respectively. The same deduction can be made with the dissimilarity initialisation. Since Scenario *B* and *C* are the ones evaluating the text contribution to the clustering, we advise not to use pre-trained embeddings.

Finally, the best ARI are all obtained with the dissimilarity initialisation. Consequently, Deep-LPTM will only be initialised with it in the rest of the paper.

### 4.4 | Model selection

In order to assess the model selection criterion, this section provides two experiments. First, we evaluate IC2L relevancy to select  $Q$  on all three scenarios. Second, we test IC2L efficiency to select the triplet  $(K, P, Q)$  on Scenario *C* specifically.

#### 4.4.0.1 | Selection of $Q$ with $P = 2$ and the true $K$

Keeping  $P$  set to 2 and  $K$  fixed to its true value for the moment, Table 4 assesses the effectiveness of IC2L to select the number of clusters  $Q$ . In all three scenarios, IC2L selects the true model 10 times out of 10.

	Scenario A $Q^* = 3$	Scenario B $Q^* = 2$	Scenario C $Q^* = 4$
$Q = 2$	0	10	0
$Q = 3$	10	0	0
$Q = 4$	0	0	10
$Q = 5$	0	0	0
$Q = 10$	0	0	0

**TABLE 4** Number of times a value  $Q$  is selected by the IC2L criterion over 10 graphs with the true value of  $K$  and  $P = 2$ .

	$K = 2$	$K = 3$	$K = 4$	$K = 5$	$K = 6$
$Q = 2$	0	0	0	0	0
$Q = 3$	0	0	0	0	0
$Q = 4$	0	10	0	0	0
$Q = 5$	0	0	0	0	0
$Q = 6$	0	0	0	0	0

**TABLE 5** Number of times a triplet  $(K, P, Q)$  is associated with the highest IC2L over 10 graphs simulated according to Scenario C ( $Q^* = 4$  and  $K^* = 3$ ). All the models with the highest IC2L value correspond to  $P = 2$ . Therefore, only the table corresponding to this value is shown.

#### 4.4.0.2 | Selection of the triplet $(P, K, Q)$

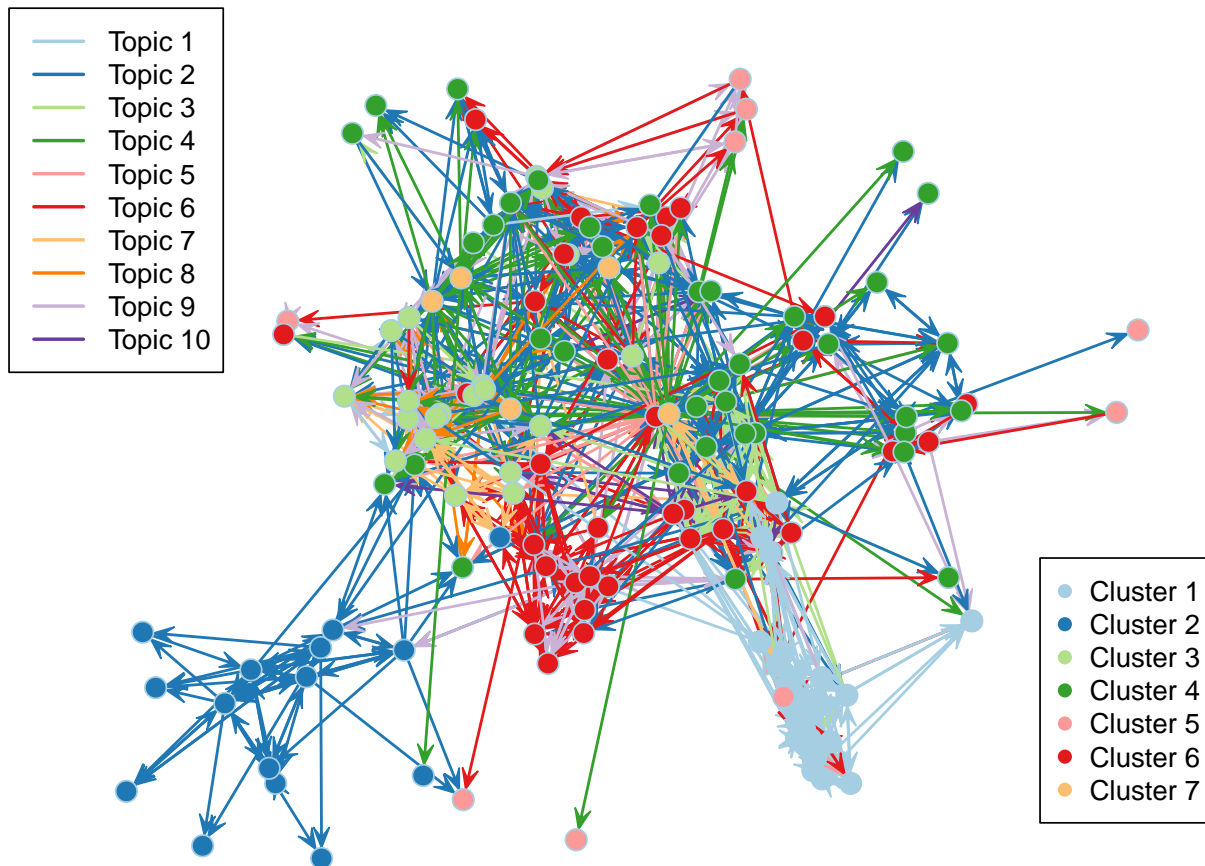
Let us now consider selecting the triplet  $(K, P, Q)$  simultaneously. Table 5 displays the number of times a triplet is selected by IC2L for 10 graphs simulated according to Scenario C (with  $Q^* = 4$  and  $K^* = 3$  the true values). The selected node embedding dimension is always  $P = 2$ , thus, we only provide  $K$  and  $Q$  in the table. First, it is satisfactory that IC2L always selects the true number of topics and clusters. Second, by always picking the dimension  $P = 2$ , IC2L favours models with a lower complexity. In our case, this translates into choosing models able to directly visualise the data in two dimensions, simply by plotting the latent vectors  $Z_i$ . This suits our purpose of building an explainable model.

## 4.5 | Benchmark

	ScenarioA	ScenarioB	ScenarioC	
Easy	SBM	1.00 ± 0.00	-0.00 ± 0.01	0.73 ± 0.05
	STBM	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.01
	ETSBM	0.99 ± 0.03	1.00 ± 0.00	0.96 ± 0.04
	ETSBM - PT	1.00 ± 0.00	1.00 ± 0.00	0.96 ± 0.05
	Deep-LPTM	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Deep-LPTM - PT	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Hard	SBM	0.97 ± 0.03	0.00 ± 0.00	0.62 ± 0.1
	STBM	0.63 ± 0.23	1.00 ± 0.00	0.66 ± 0.19
	ETSBM	0.96 ± 0.10	0.90 ± 0.30	0.72 ± 0.25
	ETSBM - PT	0.99 ± 0.01	1.00 ± 0.00	0.74 ± 0.21
	Deep-LPTM	0.99 ± 0.02	1.00 ± 0.00	0.89 ± 0.15
	Deep-LPTM - PT	1.00 ± 0.01	1.00 ± 0.00	0.85 ± 0.18

**TABLE 6** ARI of the node clustering averaged over 10 graphs in all three scenarios for the two levels of difficulty Easy and Hard. Deep-LPTM, as well as ETSBM, are presented with and without pre-trained embeddings (denoted PT). Moreover, STBM and SBM are also provided as baselines.





**FIGURE 7** Deep-LPTM representation of Enron email network. The nodes positions, the node cluster memberships (denoted by the colour of the nodes) as well as the majority topic in the documents (denoted by the colour of the edges) are estimated by Deep-LPTM.

We close this section with a benchmark study presented in Table 6 to compare Deep-LPTM with state-of-the-art ETSBM and STBM. We also provide SBM and Deep-LPM as baselines even though they are not able to take into account the text edges. As in the previous sections, the table presents the average of the ARI over 10 graphs. Each graph result is obtained by running each method with five different initialisations and by taking the one resulting in the highest ELBO. The table is presented for three different models, namely STBM, ETSBM and Deep-LPTM. The last two models are evaluated with and without pre-trained embedding. In all cases, Deep-LPTM is either as good as or better than other models. In particular, in Scenario *C* with difficulty *Hard*, the ARI of Deep-LPTM node clustering is higher than all other methods, by at least 0.15. Likewise, in Scenario *A* with difficulty *Hard*, Deep-LPTM always recover the true partition while STBM only reaches an ARI of  $0.66 \pm 0.18$ .

## 5 | APPLICATION TO THE ANALYSIS OF THE ENRON EMAIL NETWORK

Enron, formed in 1985, was an American company selling natural gas in North America. In 2001, the securities and exchange commission (SEC) opened an investigation on October, 31th, for fraud, while on August, 14th, the company was "probably in the strongest and best shape that it [had] probably ever been in" according to its CEO. On early December the same year, the company filed for the largest bankruptcy at that time. We propose here to concentrate on the critical period September, 1st to December, 31th leading to the downfall of the company in order to understand the organisation of the company during this crucial period. Thanks to the decision of the federal energy regulatory commission (FERC), the dataset is publicly accessible, and contains all 20,940 emails exchanged between 149 employees. All edges holding multiple messages were coerced into a single meta-message by stacking the documents together. As a result, the network holds 1,234 edges between the 149 employees. The dataset can be found at <https://www.cs.cmu.edu/~.enron/>.

<b>Topic 1</b> tw watson hayslett donoho lindy geaccone lynn transwestern teb lohman	<b>Topic 2</b> ercot vepco ene liz dyn filename mws desk mw enpower	<b>Topic 3</b> rto steffes christi nicolay novosel affairs rtos shapiro government skilling	<b>Topic 4</b> backup seat location test supplies building floors mails notified seats	<b>Topic 5</b> ofo interview cycle mmbtu usage interviewers fantastic super deliveries dinner
<b>Topic 6</b> sara shackleton kim ward master isda perlingiere perlingiereenron leathercenter shackletonenron	<b>Topic 7</b> frontier western williams dt project whitt dth enw marathon cheyenne	<b>Topic 8</b> grigsby desk mike taleban forces sheppard afghanistan holst gaskill ina	<b>Topic 9</b> master nymex handling isda executed agreement netting multicurrency na cn	<b>Topic 10</b> edison puc dwr davis dasovich sce da state california jeff

**FIGURE 8** The 10 most probable words of each topic according to Deep-LPTM.

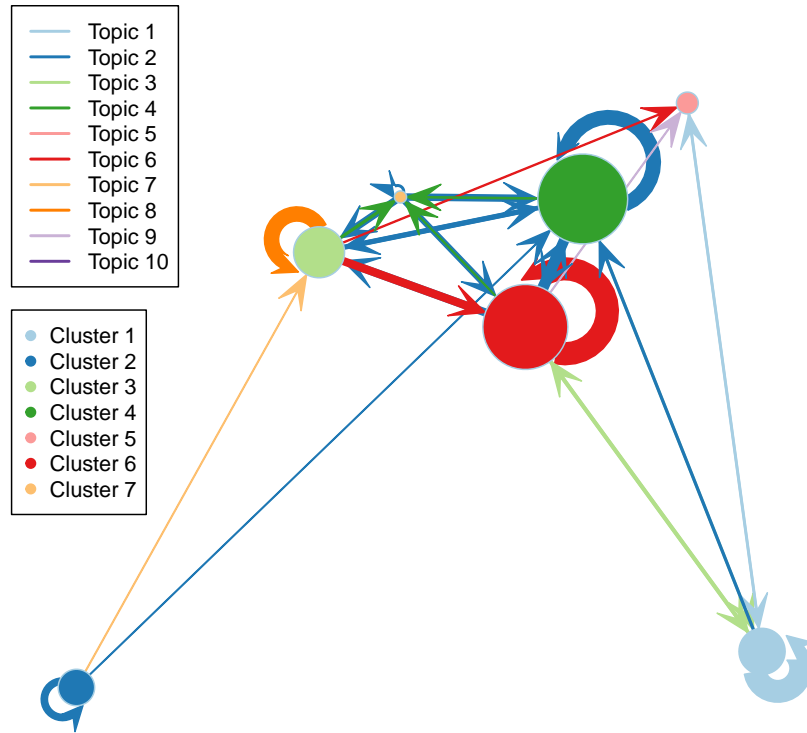
The estimation of Deep-LPTM is conducted for all triplets  $(Q, K, P)$  where  $Q \in \{5, 7, 10\}$ ,  $K \in \{3, 5, 7, 10\}$  and  $P \in \{2, 4, 8, 16\}$ . The highest value of IC2L corresponds to the triplet  $(Q, K, P) = (7, 10, 2)$ . Deep-LPTM clustering results are displayed in Figure 7. The node cluster memberships as well as the edge majority topics are represented by their respective colours. In addition, the topics are interpreted by looking at their corresponding most probable words in Figure 8. Interestingly, IC2L selected a low-dimensional node latent space, which fits with the observations made in [4].

#### 5.0.0.1 | Topics analysis

The topics can be depicted as follow:

- Topic 1 concerns Charles Watson, Dynegy CEO at the time, that negotiated a deal to finance Enron, involving the *transwestern* pipeline, and to merge the companies
- Topic 2 refers to regional energy
- Topic 3 deals with business operations
- Topic 4 is related to office supplies and day-to-day work
- Topic 5 mentions the energy usage and delivery
- Topic 6 is related to legal and strategical aspects of Enron business, involving Sara Shakleton (vice president of Enron North America Corporation), and Debra Perlingiere, from the legal department
- Topic 7 is concerned with infrastructures and geographical projects
- Topic 8 corresponds to a discussions about Enron activities in Afghanistan, which may be seen as sensitive given the american situation in 2001
- Topic 9 focuses on financial aspects
- Topic 10 mentions the California electricity crisis, which almost led to the bankruptcy of the Southern California Edison corporation

The topics as well as the visualisation provide significant information on the dataset. In particular, Deep-LPTM identifies different departments and cases of the company through the topics and successfully represent it in the graph structure as we shall detail.



**FIGURE 9** Enron email meta-network, based on Deep-LPTM estimates, represents the clusters and their interactions. The node sizes depend on the number of email accounts assigned to each cluster and the node positions are estimated by Deep-LPTM. The colours of the edges refer to the majority topic between the respective clusters and the sizes of the edges depend on the number of connections between the clusters. We keep only the edges corresponding to more than five connections for readability purposes.

#### 5.0.0.2 | Cluster analysis

The meta-network in Figure 9 gives an overview of the results to analyse the clusters. We keep only the edges corresponding to more than five connections for readability purposes. However, important details provided by Figure 7 could not be uncovered with the meta-network only.

First, Clusters 1 and 2 are well separated from the rest of the graph. Moreover, each one is characterised by a specific topic. Cluster 1 is involved in discussions about a financial deal and a possible merge (Topic 1) and more than half of the employees in Cluster 1 have a managerial position or work in the legal department (vice presidents, directors, lawyer). Cluster 2 refers to the regional energy business (Topic 2) and nine out of the 15 nodes correspond to either employees or a traders.

Second, Cluster 6 displays a high level of internal connectivity and is essentially featured by discussions involving the legal department (Topic 6) as well as financial aspects of the company (Topic 9). The status of the employees in that cluster are mainly composed of directors, vice presidents as well as presidents of the company. Interestingly, Deep-LPTM placed a president, corresponding to the node with the highest degree in the graph, at the centre of the network, and allocated it to the cluster of managers of the company. This graphical property, unique to Deep-LPTM, stresses the incorporation of the connections, as well as the topics in the emails, to obtain a meaningful representation of the network. Conversely, in Figure 9, the meta-network is not able to dissociate the connectivity of a node with a different connectivity than the rest of the cluster. Thus, Cluster 6 is central as a whole in the network in Figure 9.

Third, the topics involved in emails of Clusters 3 and 4 are the main drivers of the characterisation of the two clusters. For instance, Cluster 3 is highly connected to the graph and is involved in discussions about infrastructures and geographical aspects (Topic 7), as well as financial aspects (Topic 9). Emails related to Afghanistan were exchanged, mainly involving nodes from Cluster 3. It is worth noticing that nodes in cluster 3 correspond to people with a high position in the hierarchy of the company (almost two third of the employees in the clusters are either managers, directors, vice presidents, presidents or CEO). In addition,

although many nodes in Cluster 4 are involved in regional energy discussions (Topic 2) like Cluster 2, they also discuss about day to day business questions (Topic 4), and are highly connected with the rest of the network, unlike Cluster 2.

Finally, Cluster 7 is composed of four nodes that are at the crossroads between several clusters and topics. We close this analysis with Cluster 5 that is composed of nodes mainly receipting emails and poorly connected to the graph, highlighted in the representation by the node positions being on the rim of the network.

## 6 | CONCLUSION

We introduced a novel deep probabilistic model, named the deep latent position topic model (Deep-LPTM) to analyse networks with textual edges. Deep-LPTM allows to simultaneously cluster nodes, to model the topics in the documents, and to provide a network visualisation. To represent networks into a Euclidean space, most methodologies rely either on heuristic method used a posteriori or only focus on the connectivity of the graph. On the contrary, this work tackles this problem by incorporating the network representation into the modelling, enabling the clustering as well as the topic modelling to be included in the calculation of the node positions. To benefit from the flexibility of deep neural networks, our methodology is based on a graph convolutional network (GCN) to encode the nodes into a vector space using the connectivity of the graph as well as a neural topic model to encode the documents and the topics into a vector space. Even though we focused on directed networks, the extension to undirected networks is straightforward. The applications of this methodology are numerous, including social sciences, journalism or social network analysis. The proposed methodology relies on a variational inference algorithm to maximise the marginal likelihood. The optimisation combines analytical formulas and stochastic gradient descent steps to estimate the parameters. In this paper, we also derived the integrated classification and latent likelihood (IC2L) criterion to choose relevant numbers of clusters and topics as well as the dimension of the node latent space. Both the extensive benchmark study as well as the Enron emails analysis highlighted the visualisation power and the clustering efficiency of the proposed methodology. In future work, it is crucial to adapt this methodology to networks with multiple edges, corresponding to the exchange of several documents between nodes.

□

## APPENDIX

### A GRAPHICAL MODEL

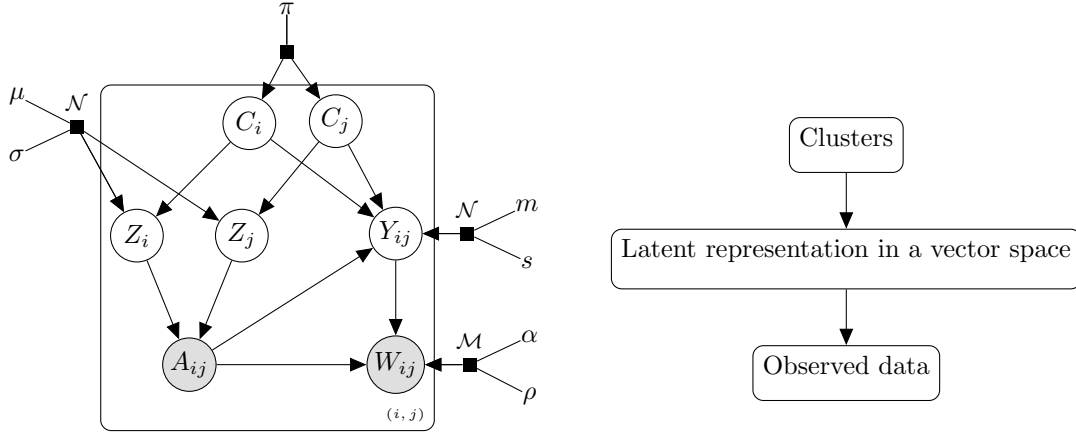
Figure A1 provides the graphical representation of the model with its parameters.

### B COMPUTATION OF THE ELBO TERMS

In this section, computational details regarding the ELBO are provided term by term. First,  $\mathbb{E}_R [\log p(A | Z, \kappa)]$  is given by:

$$\mathbb{E}_R [\log p(A | Z, \kappa)] = \sum_{i,j=1}^N \{A_{ij} \mathbb{E}_R [\log p_{ij}] + (1 - A_{ij}) \mathbb{E}_R [\log(1 - p_{ij})]\},$$

where  $p_{ij} = (1 + e^{-\eta_{ij}})^{-1}$  and  $\eta_{ij} := \kappa - \|Z_i - Z_j\|$ .



**FIGURE A1** Graphical model with the parameters of Deep-LPTM where the  $Z_i$ s denote the latent node vectors,  $Y_{ij}$ s the latent document vector,  $C_i$ s the node cluster memberships,  $A = (A_{ij})_{ij} \in \mathcal{M}_{N \times N}(\{0, 1\})$  the binary adjacency matrix and  $W_{ij}$  the document sent by node  $i$  to node  $j$ .

Second, denoting  $\beta_k = \text{softmax}(\rho^\top \alpha_k) \in \mathbb{R}^V$ ,  $\beta = (\beta_1 \dots \beta_K)^\top \in \mathcal{M}_{K \times V}(\mathbb{R})$  and  $w_{ij}^v = (\beta^\top \tilde{Y}_{ij})_v$ , the probability for the word  $v$  to appear in document  $W_{ij}$ , for any  $v \in \{1, \dots, V\}$ ,  $\mathbb{E}_R [\log p(W \mid A, Y, \rho, \alpha)]$  is:

$$\begin{aligned} \mathbb{E}_R [\log p(W \mid A, Y, \rho, \alpha)] &= \sum_{i,j=1}^N A_{ij} \mathbb{E}_R [\log \mathcal{M}_V (W_{ij}; M_{ij}, \beta^\top \text{softmax}(Y_{ij}))] \\ &= \sum_{i,j} A_{ij} \mathbb{E}_R \left[ \log \frac{M_{ij}!}{\prod_{v=1}^V (W_{ij}^v)!} \prod_{v=1}^V (w_{ij}^v)^{W_{ij}^v} \right]. \end{aligned}$$

The difference between the terms related to the cluster memberships,  $\mathbb{E}_R [\log p(C \mid \pi)]$  and  $\mathbb{E}_R [\log R(C)]$ , gives the following:

$$\begin{aligned} \mathbb{E}_R [\log p(C \mid \pi)] - \mathbb{E}_R [\log R(C)] &= \sum_{i=1}^M \sum_{q=1}^Q \mathbb{E}_R [C_{iq} \log \pi_q] - \mathbb{E}_R [C_{iq} \log \tau_{iq}] \\ &= \sum_{i=1}^M \sum_{q=1}^Q \tau_{iq} \log \frac{\pi_q}{\tau_{iq}}. \end{aligned}$$

The difference between the generative distribution of the node positions term  $\mathbb{E}_R [\log p(Z | C, \mu, \sigma)]$  and the one related to the variational distribution of the node positions  $\mathbb{E}_R [\log R(Z | A)]$  gives:

$$\begin{aligned} & \mathbb{E}_R [\log p(Z | C, \mu, \sigma)] - \mathbb{E}_R [\log R(Z | A)] \\ &= \sum_{i=1}^N \sum_{q=1}^Q \mathbb{E}_R [C_{iq} \log \mathcal{N}(Z_i; \mu_q, \sigma_q^2 I_p)] \end{aligned} \quad (\text{B1})$$

$$\begin{aligned} & - \sum_{i=1}^N \mathbb{E}_R [\log \mathcal{N}(Z_i; \mu_{\phi_Z}(A)_i, \sigma_{\phi_Z}^2(A)_i I_p)] \\ &= - \sum_{i=1}^N \sum_{q=1}^Q \tau_{iq} \text{KL}(\mathcal{N}(\mu_{\phi_Z}(A)_i, \sigma_{\phi_Z}^2(A)_i I_p) \parallel \mathcal{N}(\mu_q, \sigma_q^2 I_p)) \\ &= - \sum_{i=1}^N \sum_{q=1}^Q \tau_{iq} \underbrace{\left[ \log \frac{\sigma_q^p}{\sigma_{\phi_Z}(A)_i^p} - \frac{p}{2} + \frac{p\sigma_{\phi_Z}^2(A)_i + \|\mu_{\phi_Z}(A)_i - \mu_q\|_2^2}{2\sigma_q^2} \right]}_{\text{KL}_{iq}^Z(\mu_{\phi_Z}(A)_i, \sigma_{\phi_Z}(A)_i, \mu_q, \sigma_q)} \\ &= - \sum_{i=1}^N \sum_{q=1}^Q \tau_{iq} \text{KL}_{iq}^Z(\mu_{\phi_Z}(A)_i, \sigma_{\phi_Z}(A)_i, \mu_q, \sigma_q). \end{aligned} \quad (\text{B2})$$

Symmetrically, the term regarding the edge positions is obtained as follow:

$$\begin{aligned} & \mathbb{E}_R [\log p(Y | A, C, m, s)] - \mathbb{E}_R [\log R(Y | A, W)] \\ &= \sum_{i,j=1}^N \sum_{q,r=1}^Q \mathbb{E}_R [A_{ij} C_{iq} C_{jr} \log \mathcal{N}(Y_{ij}; m_{qr}, s_{qr}^2 I_K)] \\ & - \sum_{i,j=1}^N \mathbb{E}_R [A_{ij} \log \mathcal{N}(Y_{ij}; \mu_{\phi_Y}(W_{ij}), \text{diag}(\sigma_{\phi_Y}^2(W_{ij})))] \\ &= - \sum_{i,j=1}^N \sum_{q,r=1}^Q A_{ij} \tau_{iq} \tau_{jr} \text{KL}(\mathcal{N}(\mu_{\phi_Y}(W_{ij}), \text{diag}(\sigma_{\phi_Y}^2(W_{ij}))) \parallel \mathcal{N}(m_{qr}, s_{qr}^2 I_K)) \\ &= - \sum_{i,j=1}^N \sum_{q,r=1}^Q A_{ij} \tau_{iq} \tau_{jr} \text{KL}_{ijqr}^Y(\mu_{\phi_Y}(W_{ij}), \sigma_{\phi_Y}(W_{ij}), m_{qr}, s_{qr}), \end{aligned}$$

where

$$\begin{aligned} & \text{KL}_{ijqr}^Y(\mu_{\phi_Y}(W_{ij}), \sigma_{\phi_Y}(W_{ij}), m_{qr}, s_{qr}) \\ &= K \log s_{qr} - \sum_{k=1}^K \log \sigma_{\phi_Y}(W_{ij})_k - \frac{K}{2} \\ & + \frac{\sum_{k=1}^K \sigma_{\phi_Y}^2(W_{ij})_k + \|\mu_{\phi_Y}(W_{ij}) - m_{qr}\|_2^2}{2s_{qr}^2}. \end{aligned}$$

## C INFERENCE

### C.1 Optimisation

In this section, we provide the optimisation steps to recover the parameters maximising the ELBO. To begin with, let us recall the ELBO expression:

$$\begin{aligned}
\mathcal{L}(\tau, \phi_Z, \phi_Y; \pi, \mu, \sigma, \kappa, m, s, \alpha, \rho) &:= \mathcal{L}(R(\cdot), \Theta) \\
&= \sum_{i,j} \{A_{ij} \mathbb{E}_R [\log p_{ij}] + (1 - A_{ij}) \mathbb{E}_R [\log(1 - p_{ij})]\} \\
&\quad + \sum_{i,j} A_{ij} \mathbb{E}_R \left[ \log \frac{M_{ij}!}{\prod_{v=1}^V (W_{ij}^v)!} \prod_{v=1}^V (w_{ij}^v)^{W_{ij}^v} \right] \\
&\quad - \sum_{i=1}^N \sum_{q=1}^Q \tau_{iq} \text{KL}_{iq}^Z (\mu_{\phi_Z}(A)_i, \sigma_{\phi_Z}^2(A)_i I_p, \mu_q, \sigma_q^2 I_p) \\
&\quad - \sum_{i,j=1}^N \sum_{q,r=1}^Q A_{ij} \tau_{iq} \tau_{jr} \text{KL}_{ijqr}^Y (\mu_{\phi_Y}(W_{ij}), \text{diag}(\sigma_{\phi_Y}^2(W_{ij})), m_{qr}, s_{qr}^2 I_K) \\
&\quad + \sum_{i=1}^N \sum_{q=1}^Q \tau_{iq} \log \frac{\pi_q}{\tau_{iq}}.
\end{aligned}$$

#### C.1.0.1 Update of $\tau$

First, we optimise the ELBO with respect to  $\tau_{iq}$ . Since  $\tau_i \in \Delta_{Q-1}$ , the term  $c_i(1 - \sum_{q=1}^Q \tau_{iq})$  is added to the ELBO, giving the Lagrangian of the function. Thus, the derivative of the Lagrangian with respect to  $\tau_{iq}$  gives:

$$\begin{aligned}
\frac{\partial}{\partial \tau_{iq}} \mathcal{L}(R(\cdot); \Theta) &= -\text{KL}_{iq}^Z - \sum_{j=1}^N \sum_{r=1}^Q \{A_{ij} \tau_{jr} \text{KL}_{ijqr}^Y + A_{ji} \tau_{jr} \text{KL}_{jirq}^Y\} \\
&\quad + \log \frac{\pi_q}{\tau_{iq}} - 1 - c_i.
\end{aligned}$$

Setting this partial derivative to zero gives:

$$\log \tau_{iq} = \underbrace{-\text{KL}_{iq}^Z - \sum_{j=1}^N \sum_{r=1}^Q \{A_{ij} \tau_{jr} \text{KL}_{ijqr}^Y + A_{ji} \tau_{jr} \text{KL}_{jirq}^Y\}}_{T_{iq}} + \log \pi_q - 1 - c_i.$$

Thus  $\tau_{iq} = e^{T_{iq}} e^{-c_i}$ . Moreover, since  $\sum_{q=1}^Q \tau_{iq} = 1$ , we have  $e^{c_i} = \sum_{q=1}^Q e^{T_{iq}}$ . Therefore,  $\tau_{iq} = e^{T_{iq}} / (\sum_{q=1}^Q e^{T_{iq}})$ . The complete form is:

$$\tau_{iq} = \frac{\pi_q e^{-\text{KL}_{iq}^Z - \sum_{j \neq i} \sum_{r=1}^Q (A_{ij} \tau_{jr} \text{KL}_{ijqr}^Y + A_{ji} \tau_{jr} \text{KL}_{jirq}^Y)}}{\sum_{l=1}^Q \pi_l e^{-\text{KL}_{il}^Z - \sum_{j \neq i} \sum_{r=1}^Q (A_{ij} \tau_{jr} \text{KL}_{ijqr}^Y + A_{ji} \tau_{jr} \text{KL}_{jirq}^Y)}}. \tag{C3}$$

#### C.1.0.2 Update of $\pi$

Since  $\pi \in \Delta_{Q-1}$ , this constraint is added to the function to obtain the lagrangian. This corresponds to adding the term  $c(1 - \sum_{q=1}^Q \pi_q)$  to the ELBO. Thus, the partial derivative of the Lagrangian  $\mathcal{L}$  with respect to  $\pi_q$  of the lagrangian is:

$$\frac{\partial}{\partial \pi_q} \mathcal{L}(R(\cdot); \Theta, c) = \sum_{i=1}^N \frac{\tau_{iq}}{\pi_q} - c.$$

Setting this to zero gives:

$$\frac{1}{\pi_q} \sum_{i=1}^N \tau_{iq} = c.$$

Multiplying by  $\pi_q$  and summing over  $q$  gives that  $c = N$ . Therefore, after plugging it back into the previous expression, the following holds:

$$\pi_q = \frac{1}{N} \sum_{i=1}^N \tau_{iq}. \quad (C4)$$

### C.1.0.3 Updates of $\mu_q$ and $\sigma_q$

Taking the partial derivative of the ELBO with respect to  $\mu_q$  gives the following:

$$\frac{\partial}{\partial \mu_q} \mathcal{L}(R(\cdot); \Theta) = - \sum_{i=1}^N \frac{\tau_{iq}}{2\sigma_q^2} (2\mu_q - 2\mu_{\phi_Z}(A)_i).$$

Therefore, setting this quantity to zero gives the following update for  $\mu_q$ :

$$\mu_q = \left( \sum_{i=1}^N \tau_{iq} \right)^{-1} \sum_{i=1}^N \tau_{iq} \mu_{\phi_Z}(A)_i. \quad (C5)$$

The partial derivate of the ELBO with respect to  $\sigma_q$  is:

$$\frac{\partial}{\partial \sigma_q} \mathcal{L}(R(\cdot); \Theta) = - \sum_{i=1}^N \tau_{iq} \left( \frac{p}{\sigma_q} - \frac{p\sigma_{\phi_Z}^2(A)_i + \|\mu_{\phi_Z}(A)_i - \mu_q\|_2^2}{2} \frac{2\sigma_q}{\sigma_q^4} \right).$$

Thus, the first order condition on  $\sigma_q$  gives the following:

$$\begin{aligned} \frac{p}{\sigma_q} \sum_{i=1}^N \tau_{iq} &= \frac{1}{\sigma_q^3} \sum_{i=1}^N \tau_{iq} (p\sigma_{\phi_Z}^2(A)_i + \|\mu_{\phi_Z}(A)_i - \mu_q\|_2^2) \\ \sigma_q^2 &= \left( p \sum_{i=1}^N \tau_{iq} \right)^{-1} \sum_{i=1}^N \tau_{iq} (p\sigma_{\phi_Z}^2(A)_i + \|\mu_{\phi_Z}(A)_i - \mu_q\|_2^2). \end{aligned} \quad (C6)$$

### C.1.0.4 Updates of $m$ and $s$

As in the previous sections, we optimise the ELBO with respect to  $m$  and  $s$  with the first order conditions. The partial derivate of  $\mathcal{L}$  with respect to  $m_{qr}$  is:

$$\frac{\partial}{\partial m_{qr}} \mathcal{L}(R(\cdot); \Theta) = - \frac{1}{2s_{qr}^2} \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr} (2m_{qr} - 2\mu_{\phi_Y}(W_{ij})).$$

Therefore, setting this expression to zero gives the following update for  $m_{qr}$ :

$$m_{qr} = \left( \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr} \right)^{-1} \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr} \mu_{\phi_Y}(W_{ij}). \quad (C7)$$

The partial derivate of the ELBO with respect to  $s_{qr}$  is:

$$\begin{aligned} \frac{\partial}{\partial s_{qr}} \mathcal{L}(R(\cdot); \Theta) &= - \sum_{i=1}^N A_{ij} \tau_{iq} \tau_{jr} \left( \frac{K}{s_{qr}} \right. \\ &\quad \left. - \frac{\sum_{k=1}^K \sigma_{\phi_Y}^2(W_{ij})_k + \|\mu_{\phi_Y}(W_{ij}) - m_{qr}\|_2^2}{s_{qr}^3} \right). \end{aligned}$$



Thus, the first order condition on  $s_{qr}$  gives the following:

$$s_{qr}^2 K \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr} = \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr} \left[ \sum_{k=1}^K \sigma_{\phi_Y}^2 (W_{ij})_k + \|\mu_{\phi_Y}(W_{ij}) - m_{qr}\|_2^2 \right].$$

Hence, the update of  $s_{qr}^2$  is given by:

$$s_{qr}^2 = \left( K \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr} \right)^{-1} \sum_{i,j=1}^N A_{ij} \tau_{iq} \tau_{jr} \left[ \sum_{k=1}^K \sigma_{\phi_Y}^2 (W_{ij})_k + \|\mu_{\phi_Y}(W_{ij}) - m_{qr}\|_2^2 \right].$$

## C.2 Derivation of the selection model criterion

*Proof of Proposition 3.* Assuming a fully factorised prior distribution  $p(\kappa, \pi, \mu, \sigma, m, s, \rho, \alpha) = p(\kappa)p(\pi)p(\mu)p(\sigma)p(m)p(s)p(\rho)p(\alpha)$ , Lemma 3.1 in [40] can be directly extended to our case to decompose the integral in (24):

$$\begin{aligned} \log p(A, W, Z, Y, C \mid \mathcal{M}, Q, K, P) &= \log p(A \mid Z, \mathcal{M}) + \log p(Z \mid C, \mathcal{M}, Q, P) \\ &\quad + \log p(W \mid A, Y, \mathcal{M}) + \log p(Y \mid A, C, \mathcal{M}, K) \\ &\quad + \log p(C \mid \mathcal{M}, Q). \end{aligned} \tag{C8}$$

Unfortunately, this expression cannot be computed since each term requires an integral with respect to the corresponding parameter. For instance,  $p(A \mid Z, \mathcal{M})$  cannot be integrated analytically because of the logistic link function. Fortunately, a BIC-like approximation can be derived for  $p(A \mid Z, \mathcal{M})$ ,  $p(Z \mid C, \mathcal{M}, Q, P)$ ,  $p(Y \mid A, C, \mathcal{M}, K)$  and  $p(W \mid A, Y, \mathcal{M})$ . For instance,  $p(A \mid Z, \mathcal{M})$  can be approximated by:

$$\begin{aligned} \log p(A \mid Z, \mathcal{M}) &= \log \int_{\kappa} p(A \mid \kappa, Z, \mathcal{M}) p(\kappa) d\kappa \\ &\approx \max_{\kappa} \log p(A \mid Z, \kappa, \mathcal{M}) - \frac{\nu_{A, \mathcal{M}}}{2} \log(n_A), \end{aligned}$$

where  $\nu_{A, \mathcal{M}} = 1$  denotes the number of free components in  $\kappa$  and  $n_A = N(N-1)$  denotes the number of observations in  $A$ . This can be applied to all terms except  $p(C \mid \mathcal{M}, Q)$  since the posterior cluster memberships probabilities  $\tau_i$  can be on the boundary of the parameter space. Fortunately, this term can be computed analytically. By assuming a Dirichlet prior  $\mathcal{D}_Q(\delta_1, \dots, \delta_Q)$  on the topic proportions  $\pi$ :

$$\begin{aligned} p(C \mid \mathcal{M}, Q) &= \int p(C \mid \pi, \mathcal{M}, Q) p(\pi) d\pi \\ &= \frac{\Gamma\left(\sum_{q=1}^Q \delta_q\right) \prod_{q=1}^Q \Gamma(n_q + \delta_q)}{\prod_{q=1}^Q \Gamma(\delta_q) \Gamma\left(\sum_{q=1}^Q n_q + \delta_q\right)}, \end{aligned}$$

where  $n_q := \sum_{i=1}^N C_{iq}$ . In this paper, we consider the non informative Jeffreys prior distribution ( $\delta_q = 1/2$ ), as in [40] and [3]. Moreover, since  $C_i$  is not available, we replace it with its maximum-a-posteriori estimate  $\hat{C}_i$  where  $\hat{C}_{iq} = 1$  if  $q = \operatorname{argmax}(\tau_{i1}, \dots, \tau_{iQ})$ , and 0 otherwise, which in turn gives  $\hat{n}_q := \sum_{i=1}^N \hat{C}_{iq}$ . Using Stirling formula to approximate the Gamma function for a large value of  $N$ , we obtain:

$$p(\hat{C} \mid \mathcal{M}, Q) \approx p(\hat{C} \mid \hat{\pi}, \mathcal{M}, Q) - \frac{Q-1}{2} \log(N). \tag{C9}$$

To conclude, since  $Z$  and  $Y$  are not available, we replace the missing data with their maximum-a-posteriori estimates  $\hat{Z}$  and  $\hat{Y}$ . Denoting  $IC2L$  the quantity  $\log p(A, W, \hat{Z}, \hat{Y}, \hat{C} \mid \mathcal{M}, Q, K, P)$ , we obtain:

$$\begin{aligned}
IC2L &= \max_{\theta} \log p(A, W, \hat{Z}, \hat{Y}, \hat{C} \mid \theta, \mathcal{M}, Q, K, P) - \Omega(\mathcal{M}, Q, K, P) \\
&= \max_{\kappa} \log p(A \mid \hat{Z}, \kappa, \mathcal{M}) - \frac{1}{2} \log(N(N-1)) \\
&+ \max_{\mu, \sigma} \log p(\hat{Z} \mid \hat{C}, \mu, \sigma, \mathcal{M}, Q, P) - \frac{QP + Q}{2} \log(N) \\
&+ \max_{\rho, \alpha} \log p(W \mid A, \hat{Y}, \rho, \alpha, \mathcal{M}) - \frac{VL + KL}{2} \log(M) \\
&+ \max_{m, s} \log p(\hat{Y} \mid A, \hat{C}, m, s, \mathcal{M}, K) - \frac{Q^2K + Q^2}{2} \log(M) \\
&+ \max_{\pi} \log p(\hat{C} \mid \pi, \mathcal{M}, Q) - \frac{Q-1}{2} \log(N),
\end{aligned}$$

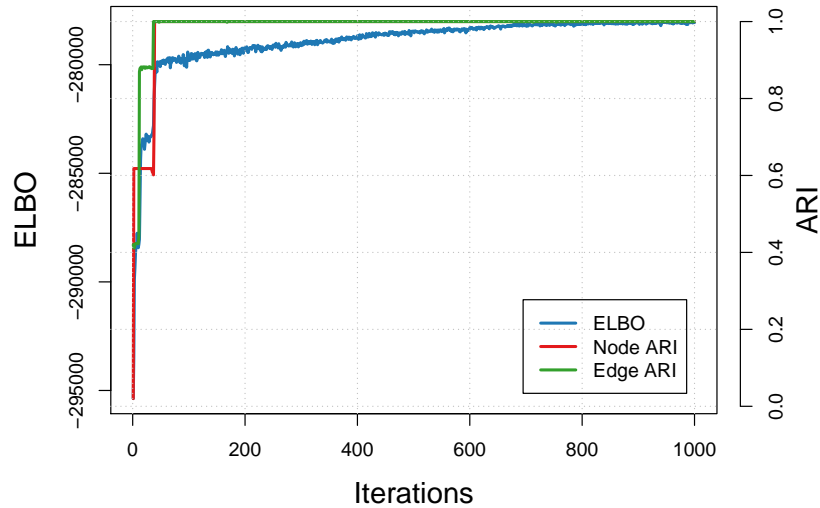
where

$$\begin{aligned}
\Omega(\mathcal{M}, Q, K, P) &= \frac{1}{2} \log(N(N-1)) \\
&+ \frac{Q(P+2) - 1}{2} \log(N) \\
&+ \frac{L(V+K) + Q^2(K+1)}{2} \log(M).
\end{aligned}$$

□

## D NUMERICAL EXPERIMENTS

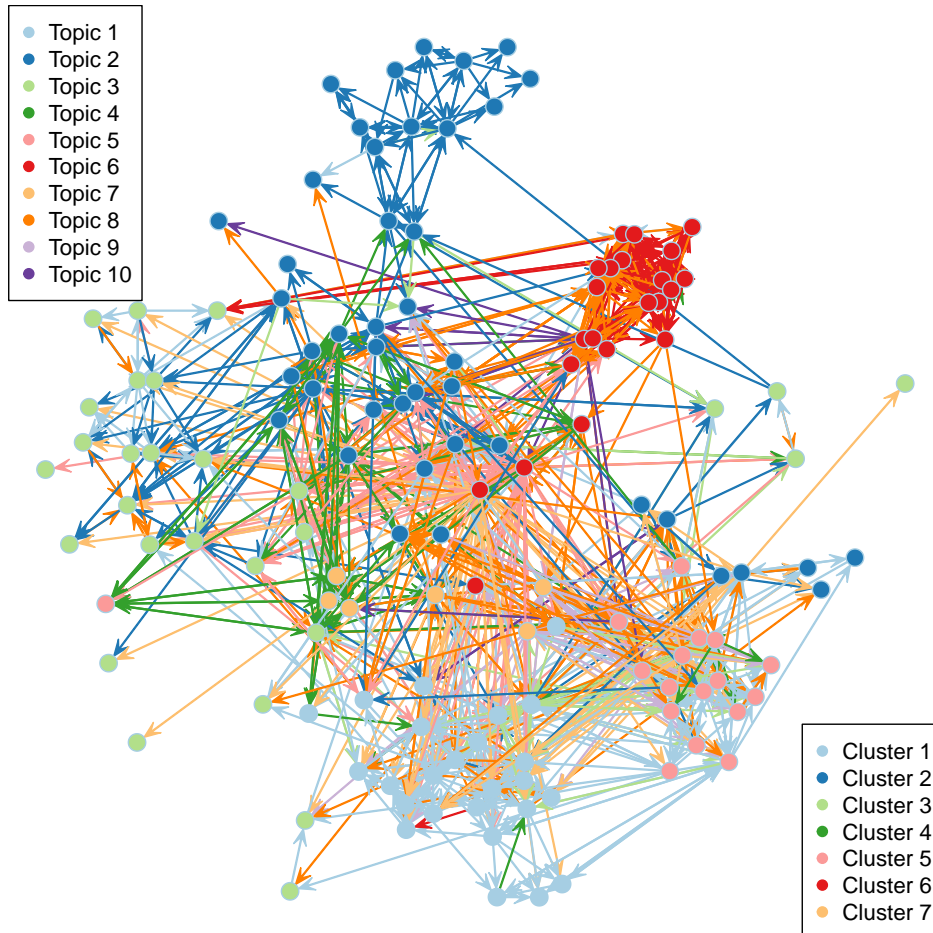
This section provides the entire ELBO evolution as well as the evolutions of the node ARI and the edge ARI, corresponding to the example in Section 4.2.



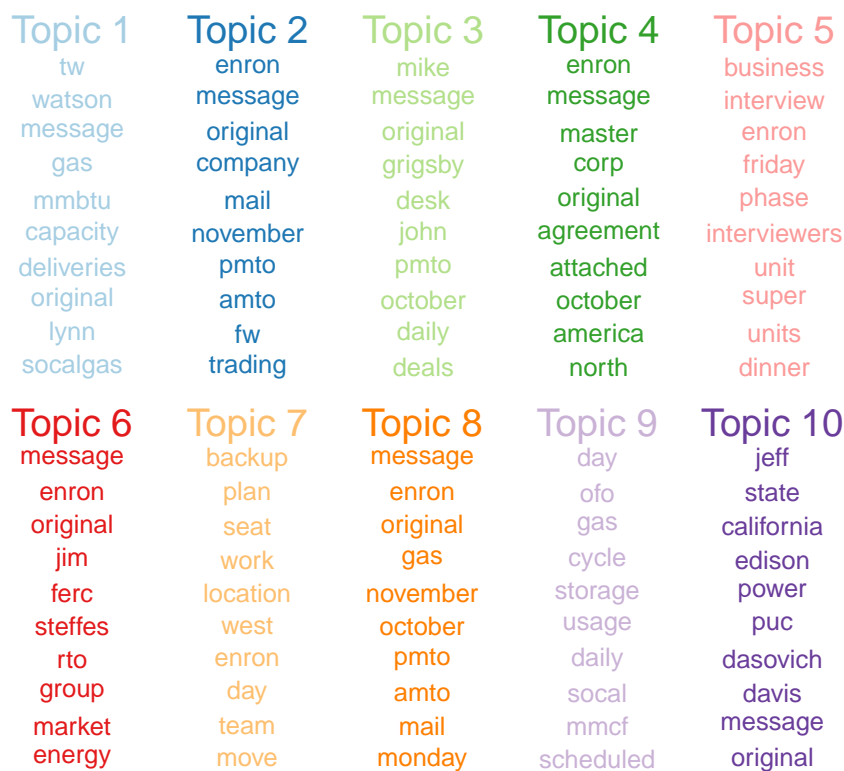
**FIGURE D2** Evolution of the ELBO, as well as the nodes ARI during the optimisation.

## E APPLICATION TO THE ANALYSIS OF THE ENRON EMAIL NETWORK

To compare our results with a state-of-the-art approach, we provide an analysis of the Enron email network with ETSBM. The node positions displayed in Figure E3 were obtained using a Fruchterman-Reingold algorithm [36] and the top words associated to ETSBM topics are provided in Figure E4.



**FIGURE E3** ETSBM clustering results on the Enron email network. The node positions were obtained using the Fruchterman-Reingold algorithm.



**FIGURE E4** The 10 most probable words of each topic according to ETSBM.

## References

- [1] Holland PW, Laskey KB, Leinhardt S. Stochastic blockmodels: First steps. *Social networks*. 1983;5(2):109–137.
- [2] Snijders TA, Nowicki K. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*. 1997;14(1):75–100.
- [3] Daudin JJ, Picard F, Robin S. A mixture model for random graphs. *Statistics and computing*. 2008;18(2):173–183.
- [4] Hoff PD, Raftery AE, Handcock MS. Latent space approaches to social network analysis. *Journal of the American Statistical Association*. 2002;97(460):1090–1098.
- [5] Handcock MS, Raftery AE, Tantrum JM. Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*. 2007;170(2):301–354.
- [6] Snijders TA. Statistical models for social networks. *Annual review of sociology*. 2011;37:131–153.
- [7] Bouveyron C, Celeux G, Murphy TB, Raftery AE. *Model-based clustering and classification for data science: with applications in R*. 50. Cambridge University Press, 2019.
- [8] Aljalbout E, Golkov V, Siddiqui Y, Strobel M, Cremers D. Clustering with deep learning: Taxonomy and new methods. 2018.
- [9] Kingma DP, Welling M. Auto-Encoding Variational Bayes. 2014.
- [10] Rezende DJ, Mohamed S, Wierstra D. Stochastic backpropagation and approximate inference in deep generative models. In: *Proceedings of Machine Learning Research*. 2014:1278–1286.
- [11] Zhang D, Yin J, Zhu X, Zhang C. Network representation learning: A survey. *IEEE transactions on Big Data*. 2018;6(1):3–28.
- [12] Kipf TN, Welling M. Variational graph auto-encoders. 2016.
- [13] Kipf TN, Welling M. Semi-Supervised Classification with Graph Convolutional Networks. In: 2017.
- [14] Henaff M, Bruna J, LeCun Y. Deep convolutional networks on graph-structured data. 2015.
- [15] Pan S, Hu R, Long G, Jiang J, Yao L, Zhang C. Adversarially Regularized Graph Autoencoder for Graph Embedding. In: *IJCAI'18*. AAAI Press 2018:2609–2615.

- [16] Mehta N, Duke LC, Rai P. Stochastic blockmodels meet graph neural networks. In: Proceedings of Machine Learning Research. 2019:4466–4474.
- [17] Latouche P, Birmelé E, Ambroise C. Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Applied Statistics*. 2011;5(1):309–336.
- [18] Liang D, Corneli M, Bouveyron C, Latouche P. Deep latent position model for node clustering in graphs. In: 2022.
- [19] Papadimitriou CH, Raghavan P, Tamaki H, Vempala S. Latent Semantic Indexing: A Probabilistic Analysis. *Journal of Computer and System Sciences*. 2000;61(2):217-235.
- [20] Hofmann T. Probabilistic latent semantic analysis. In: 1999:289–296.
- [21] Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. *the Journal of machine Learning research*. 2003;3:993–1022.
- [22] Blei DM, Lafferty J. Correlated topic models. *Advances in neural information processing systems*. 2006;18:147.
- [23] Srivastava A, Sutton C. Autoencoding Variational Inference For Topic Models. In: 2017.
- [24] Dieng AB, Ruiz FJR, Blei DM. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*. 2020;8:439–453.
- [25] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. 2013.
- [26] Zhou D, Manavoglu E, Li J, Giles CL, Zha H. Probabilistic models for discovering e-communities. In: 2006:173–182.
- [27] Rosen-Zvi M, Griffiths T, Steyvers M, Smyth P. The Author-Topic Model for Authors and Documents. In: UAI '04. AUAI Press 2004:487–494.
- [28] Pathak N, Delong C, Erickson K, Banerjee A. Social topic models for community extraction. 2008.
- [29] Liu Y, Niculescu-Mizil A, Gryc W. Topic-link LDA: joint models of topic and author community. In: 2009:665–672.
- [30] Sachan M, Contractor D, Faruque TA, Subramaniam LV. Using content and interactions for discovering communities in social networks. In: 2012:331–340.
- [31] Bouveyron C, Latouche P, Zreik R. The stochastic topic block model for the clustering of vertices in networks with textual edges. *Statistics and Computing*. 2018;28(1):11–31.
- [32] Bergé LR, Bouveyron C, Corneli M, Latouche P. The latent topic block model for the co-clustering of textual interaction data. *Computational Statistics & Data Analysis*. 2019;137:247–270.
- [33] Corneli M, Bouveyron C, Latouche P, Rossi F. The dynamic stochastic topic block model for dynamic networks with textual edges. *Statistics and Computing*. 2019;29(4):677–695.
- [34] Boutin R, Bouveyron C, Latouche P. Embedded topics in the stochastic block model. *Statistics and Computing*. 2023;33(5):1–20.
- [35] Pavel N, Krivitsky MSH, Hunter DR, Butts CT, Klumb C, Goodreau SM, Morris M. Statnet: Tools for the Statistical Modeling of Network Data.
- [36] Fruchterman TM, Reingold EM. Graph drawing by force-directed placement. *Software: Practice and experience*. 1991;21(11):1129–1164.
- [37] Kingma DP, Welling M, others . An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*. 2019;12(4):307–392.
- [38] Paszke A, Gross S, Massa F, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach H, Larochelle H, Beygelzimer A, Alché-Buc dF, Fox E, Garnett R., eds. *Advances in Neural Information Processing Systems* 32, , Curran Associates, Inc., 2019:8024–8035.
- [39] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014.
- [40] Biernacki C, Celeux G, Govaert G. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*. 2000;22(7):719–725.