Topic Modeling on Document Networks with Dirichlet Optimal Transport Barycenter

Delvin Ce Zhang and Hady W. Lauw, *Senior Member, IEEE*

Abstract—Text documents are often interconnected in a network structure, e.g., academic papers via citations, Web pages via hyperlinks. On the one hand, though Graph Neural Networks (GNNs) have shown promising ability to derive effective embeddings for such networked documents, they do not assume a latent topic structure and result in *uninterpretable* embeddings. On the other hand, topic models can infer semantically interpretable topic distributions for documents by associating each topic with a group of understandable key words. However, most topic models mainly focus on plain text within documents and fail to leverage *network structure* across documents. Network connectivity reveals topic similarity between linked documents, and modeling it could uncover meaningful semantics. Motivated by above two challenges, in this paper, we propose a GNN-based neural topic model that both captures network connectivity and derives semantically interpretable topic distributions for networked documents. For network modeling, we build the model based on the theory of Optimal Transport Barycenter, which captures network structure by allowing the topic distribution of a document to generate the content of its linked neighbors. For semantic interpretability, we extend optimal transport by incorporating semantically related words in the embedding space. Since Dirichlet prior in Latent Dirichlet Allocation successfully improves topic quality, we also analyze Dirichlet as an optimal transport prior distribution to improve topic interpretability. We design rejection sampling to simulate Dirichlet distribution. Extensive experiments on document classification, clustering, link prediction, and topic analysis verify the effectiveness of our model. Code and datasets are available at https://github.com/cezhang01/dbn.

Index Terms—Graph Neural Networks, Text Mining, Optimal Transport, Dirichlet Distribution, Document Networks

1 Introduction

While text documents are primarily expressed by words, in many cases they are also interconnected in a network structure. For example, academic papers constitute a citation network, Web pages present a hyperlink network. Graph Neural Networks (GNNs) [1] are powerful tools to derive effective low-dimensional embeddings for such networked documents, which could fulfill downstream tasks, such as document classification and link prediction. However, when dealing with text documents, we usually model a latent topic structure [2] where each document is represented by a topic distribution, and each topic is characterized by a group of understandable key words. Most previous GNNs ignore such topic structure, resulting in *uninterpretable* embeddings.

Topic model is an appealing method to uncover semantically interpretable topics that occur in a corpus. However, many models, e.g., LDA [2], deal with the plain text within each document only, without considering *network connectivity* across documents. Intuitively, two connected documents tend to share similar topics, e.g., two hyperlinked news articles tend to report similar events. Modeling network structure in addition to the textual content could discover meaningful semantics and improve topic quality.

Challenge. *First*, most GNNs [1] do not have a topic structure. Their embeddings lack semantic interpretability. However, exploring latent topics and interpreting each topic by its key words could allow us to better understand the main themes in a text corpus. *Second*, although topic models offer semantic interpretability, most works [2] focus on plain text but ignore the adjacency structure, which reveals the relationship across documents. Modeling the latter would uncover meaningful insight into latent semantics and improve topic quality. *Third*, although there exist topic models for networked documents, e.g., RTM [3], they deteriorate topic quality for shorter documents with fewer word co-occurrences, resulting in less interpretable topics and worse task performances.

Approach. Optimal Transport (OT) [4] has achieved promising performance in topic modeling [5], [6]. But no existing such method explores the modeling of document connectivity. Thus, for network structure modeling, we are motivated to develop neural topic model built on the theory of *Optimal Transport Barycenter* [7]. Unlike previous topic models that leverage a document's topic distribution to generate its own observed content, OT barycenter naturally allows the topic distribution of a document to generate the content of not only itself, but also its linked neighbors. Such mechanism matches the intuition that if two linked documents share similar topics, it is possible to use topic distribution of one document to generate the content of the other.

For semantic interpretability, we extend the cost matrix of optimal transport and incorporate *pre-trained word embeddings*, which lead to interpretable topics even when texts are short. Since Dirichlet prior in LDA [2] successfully improves topic quality, we are also motivated to investigate Dirichlet as an *optimal transport prior distribution* to further boost topic interpretability.

In this paper, we extend Variational Graph Auto-Encoder (VGAE) [1], a variant of GNNs, and integrate these modeling approaches into a unified framework, named **DBN** for **D**irichlet Optimal Transport **B**arycenter for Document **N**etworks, which captures document network connectivity, and the learned topic distributions enjoy semantic interpretability.

Contributions. Our contributions are as follows. *First*, we propose DBN, a VGAE topic model that unifies document network modeling and semantic interpretability into a joint graph neural

- *Delvin Ce Zhang is with School of Computing and Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902 E-mail: delvincezhang@gmail.com*
- *Hady W. Lauw is with School of Computing and Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902 E-mail: hadywlauw@smu.edu.sg*

network framework. *Second*, to model network structure, we propose Optimal Transport Barycenter, which induces barycentric topic distributions of documents by generating observed content of network neighbors. *Third*, for semantic interpretability, we extend the cost matrix of optimal transport by incorporating pre-trained semantically related word embeddings. We further use Dirichlet distribution as optimal transport prior to boost topic quality. We design rejection sampling to simulate Dirichlet.

2 Related Work

Graph neural networks (GNNs) [8], [9] are developed to preserve both graph connectivity and vertex features, e.g., texts. Since our work is built on Variational Graph Auto-Encoder (VGAE) [1], a variant of GNNs, we primarily review its related works. ARVGA [10] improves VGAE by adversarial generative networks. DGVAE [11] replaces Gaussian with Dirichlet prior. OT-GNN [12] uses OT to prevent oversmoothing problem. GTOT [13] employs OT for GNN pre-training. GANE-OT [14] is designed for document networks, but does not have GNN. UGNN [15] is proposed for graph classification, while ours is for unsupervised text embedding. All these models are not topic models, since embeddings are not interpretable topics. In contrast, we unify VGAE with topic modeling and offer semantic interpretability.

Topic models learn interpretable topics for documents. Vayansky and Kumar [16] produce a comprehensive survey. VAE [17] motivates the development of neural models [18]. ProdLDA [19] and DVAE [20] use Dirichlet as prior. Some topic models are designed on GNNs [21], [22]. To alleviate the sparsity in short texts, some use pre-trained word embeddings [2], [23], [24]. However, they encode content only, without network structure. We incorporate document network by OT barycenter. We also use word embeddings to define cost matrix of OT.

There are only a few topic models designed on optimal transport [4]. NSTM [6] minimizes topic and word distributions by OT distance. It does not model document adjacency. DWL [25] and OTLDA [5] apply OT in the word space only, while ours integrate OT in topic, word, and structure spaces. Others [26] leverage OT to measure the distance between the generated and the true distributions. They model plain text only and do not assume prior for topics. We propose OT barycenter to capture networks, and design Dirichlet as OT prior to improve topic quality.

Models dealing with networked documents are proposed. RTM [3] is graphical model. NRTM [27] extends VAE by a multi-layer perceptron [28] for link prediction. Adjacent-Encoder [29] modifies Auto-Encoder to capture network. GTNN [30] uses GNNs. LANTM [31] optimizes a variational objective. They suffer from word sparsity problem. On the contrary, OT benefits our models by using pre-trained word and structure embeddings to avoid word sparsity. NRTM and LANTM assume Gaussian prior, while we design Dirichlet prior to enhance topic quality.

TopIN-EF [32] uses documents' publication venues for topic modeling. On the contrary, we do not require venues to be observed. We also exclude Topic-HGNN [33], since it is designed with users and queries for recommendation, while ours are not recommendation models. MALIC [34] is not designed for document networks, but for document listwise comparisons, while this paper does not focus on document comparisons. We also distinguish our work from SwAV [35], which designs optimal transport polytope to predict the cluster assignment of one augmentation using another augmentation of the same image. In contrast, we work on text data, and each document does not have multiple different augmentations. We are also different from UniDA [36], which uses OT for domain adaptation but does not focus on texts or graphs. Though SAML-CLR [37] has both GNN and OT, it is designed with images as input for classification, while we do not work on images, and our models are unsupervised.

TABLE 1: Summary of math notations.

Notation	Description
$\mathcal{G}$	a document network
$\mathcal{D}$	a corpus of documents, $\mathcal{D} = \{\mathbf{d}_i\}_{i=1}^N$
N	number of documents in the corpus, $N = \|\mathcal{D}\|$
$\mathcal{E}$	adjacency matrix, $\mathcal{E} \in \mathbb{R}^{N\times N}$
$\mathbf{d}$	a document representation in the word space, $\mathbf{d} \in \mathbb{R}^{\|\mathcal{V}\|}$
$\mathcal{V}$	vocabulary
$\mathcal{N}(i)$	the neighbor set of document i
$\mathbf{z}_i$	topic distribution of document i, $\mathbf{z}_i \in \mathbb{R}^K$
K	number of topics
$\mathbf{C}$ and $\mathbf{C}'$	cost matrix of topic-word OT and topic-structure OT, respectively
$\mathbf{T}$ and $\mathbf{T}'$	transport plan of topic-word OT and topic-structure OT, respectively
$\mathbf{t}_k$	topic embedding of topic k
$\mathbf{e}_w$	word embedding of word w
$\boldsymbol{\varepsilon}_i$	adjacency vector or neighbor distribution of document i, $\boldsymbol{\varepsilon}_i \in \mathbb{R}^N$
$\mathbf{s}_i$	structure embedding of document i

3 Background

We introduce preliminary background knowledge to be used in our models. See Table 1 for a summary of math notations.

Let $\mathcal{G} = \{\mathcal{D}, \mathcal{E}\}$ be a document network. $\mathcal{D} = \{\mathbf{d}_i\}_{i=1}^N$ is a corpus of N documents. Each document $\mathbf{d} \in \mathbb{R}^{|\mathcal{V}|}$ is a vector in the vocabulary space $\mathcal{V}$ where each element d_w is the word count of word $w \in \mathcal{V}$ in document d. $\mathcal{E} \in \mathbb{R}^{N\times N}$ is adjacency matrix, where $e_{ij} = 1$ if there is an edge between document i and j. In this paper, we model an undirected network, i.e., $e_{ij} = e_{ji}$. We will use terms *edge* and *link* interchangeably. For a document i, its neighbors are those directly linked to i, denoted as a neighbor set $\mathcal{N}(i)$. We also consider i as its own neighbor, $i \in \mathcal{N}(i)$.

We input $\mathcal{G}$ to our models and output interpretable topic distributions for documents that preserve both text content $\mathcal{D}$ within documents and network structure $\mathcal{E}$ across documents.

Definition 3.1 (Topic Model)**.** *Given a corpus of documents $\mathcal{D}$ as input, topic model, such as LDA [2], outputs **i)** K-dimensional topic distribution $\boldsymbol{z}_i \in \Delta^K$ for each document i, and **ii)** a topic-word distribution $\boldsymbol{\Phi} \in \mathbb{R}^{K\times|\mathcal{V}|}$. K is the number of latent topics. Each row in $\boldsymbol{\Phi}$ is the distribution of a topic over the vocabulary, the key words of that topic are the highest values on that row. Thus, each topic can be **semantically interpreted** by these key words.*

Most topic models deal with texts $\mathcal{D}$ only, ignoring networks $\mathcal{E}$. In this paper, we integrate both $\mathcal{D}$ and $\mathcal{E}$ into a unified model.

Definition 3.2 (Variational Graph Auto-Encoder (VGAE))**.** *Variational Graph Auto-Encoder embeds connected documents on the network into K-dimensional embeddings $\boldsymbol{Z} \in \mathbb{R}^{|\mathcal{D}|\times K}$, which preserve both document content $\mathcal{D}$ and network connectivity $\mathcal{E}$. This encoding process is implemented by variational posterior $q(\boldsymbol{Z}|\mathcal{G})$, which is a graph convolutional network [8]. The decoding process aims to reconstruct the network structure by $p(\mathcal{E}|\boldsymbol{Z})$. Formally, VGAE aims to minimize the following objection function.*

$$\mathcal{J} = -\mathbb{E}_{q(\boldsymbol{Z}|\mathcal{G})} \log p(\mathcal{E}|\mathcal{Z}) + KL[q(\boldsymbol{Z}|\mathcal{G})||p(\boldsymbol{Z})] \tag{1}$$

where the second term, KL divergence, pushes variational posterior $q(\boldsymbol{Z}|\mathcal{G})$ to a predefined prior distribution $p(\boldsymbol{Z})$ as a regularizer. Prior $p(\boldsymbol{Z})$ is usually assumed to be a Gaussian distribution with zero mean and unit diagonal covariance, i.e., $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

VGAE is not a topic model and lacks semantic interpretability. We will extend VGAE as a topic model and design Dirichlet prior.

Definition 3.3 (Optimal Transport (OT))**.** *Optimal transport measures the distance between two probabilities. Considering two probability distributions $\boldsymbol{p} \in \Delta^{D_p}$ and $\boldsymbol{q} \in \Delta^{D_q}$ where Δ^D is a $D-1$ simplex, we define OT distance between $\boldsymbol{p}$ and $\boldsymbol{q}$ as*

$$d_{\boldsymbol{C}}(\boldsymbol{p},\boldsymbol{q}) = \min_{\boldsymbol{T}\in U(\boldsymbol{p},\boldsymbol{q})} < \boldsymbol{T},\boldsymbol{C} >= \min_{\boldsymbol{T}\in U(\boldsymbol{p},\boldsymbol{q})} \sum_{u=1}^{D_p}\sum_{v=1}^{D_q} t_{uv}c_{uv}. \quad (2)$$

$< \cdot,\cdot >$ is Frobenius dot product. $\boldsymbol{T} \in \mathbb{R}_+^{D_p\times D_q}$ is transport plan where each element t_{uv} indicates the probability to transport between p_u to q_v. $U(\boldsymbol{p},\boldsymbol{q}) = \{\boldsymbol{T} \in \mathbb{R}_+^{D_p\times D_q} | \boldsymbol{T}\boldsymbol{1}_{D_q} = \boldsymbol{p}, \boldsymbol{T}^\top \boldsymbol{1}_{D_p} = \boldsymbol{q}\}$ is the transport polytope with $\boldsymbol{p}$ and $\boldsymbol{q}$ as marginal distributions, and $\boldsymbol{1}_D \in \mathbb{R}^D$ is a vector of ones. $\boldsymbol{C} \in \mathbb{R}^{D_p\times D_q}$ is cost matrix where each element c_{uv} is the cost to transport between p_u to q_v. Given $\boldsymbol{C}$, OT distance can be computed by finding the optimal transport plan $\boldsymbol{T}^*$ and returning $d_{\boldsymbol{C}}(\boldsymbol{p},\boldsymbol{q}) =< \boldsymbol{T}^*,\boldsymbol{C} >$.

Definition 3.4 (Optimal Transport Barycenter)**.** *Given a set of distributions $\{\boldsymbol{q}_i\}_{i=1}^Q$, and weights of measures $\{a_i\}_{i=1}^Q$ where $a_i > 0$ and $\sum_{i=1}^Q a_i = 1$, OT barycenter is defined as*

$$\arg\min_{\boldsymbol{p}} \sum_{i=1}^{Q} a_i d_{\boldsymbol{C}}(\boldsymbol{p},\boldsymbol{q}_i). \quad (3)$$

Barycenter $\boldsymbol{p}$ is a notion of Fréchet mean of the set $\{\boldsymbol{q}_i\}_{i=1}^Q$.

We will use OT barycenter to capture network connectivity across documents. To alleviate word sparsity problem, we will incorporate pre-trained word embeddings into cost matrix $\mathbf{C}$.

Definition 3.5 (Rejection Sampling)**.** *Rejection sampling allows to sample data points from a relatively complex distribution, e.g., Dirichlet distribution, to enable reparameterization. Specifically, since directly sampling from a target distribution $p(\boldsymbol{z})$ is difficult, we instead seek another proposal function $f(\boldsymbol{z})$, which may not be a probability distribution, such that $f(\boldsymbol{z}) \geq p(\boldsymbol{z})$ and sampling from $f(\boldsymbol{z})$ is possible. Rejection sampling contains three steps:*

1) *Sample data point $\boldsymbol{z}_i$ from proposal function $\boldsymbol{z}_i \sim f(\boldsymbol{z})$;*
2) *Sample u_i from uniform distribution $u_i \sim \mathcal{U}(0, f(\boldsymbol{z}_i))$;*
3) *If $u_i > p(\boldsymbol{z}_i)$, then $\boldsymbol{z}_i$ is rejected as an invalid sample, otherwise $\boldsymbol{z}_i$ is retained as a sample from target $p(\boldsymbol{z})$.*

After repeating rejection sampling M times, we obtain M samples, $\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_M$, some of which are rejected. Without loss of generality, we assume the first $M' \leq M$ samples are accepted. These accepted samples $\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_{M'}$ are considered as the samples from target distribution $p(\boldsymbol{z})$ with acceptance rate $\frac{M'}{M}$. If acceptance rate is high enough, we usually accept all M samples for simplicity, though a trivial proportion should be rejected.

Since Dirichlet distribution is not a location scale family and hinders reparameterization, we will investigate rejection sampling to approximate Dirichlet and enable reparameterization.

4 Model Architecture and Analysis

Our work is built on Variational Graph Auto-Encoder [1]. We first use a graph convolutional encoder to project documents into low-dimensional topic space. For a document i,

$$\tilde{\mathbf{h}}_i^{(l)} = \mathbf{W}^{(l)}\mathbf{h}_i^{(l-1)}. \quad (4)$$

l is the l-th convolutional layer, $\mathbf{W}^{(l)}$ is learnable parameter. $\mathbf{h}_i^{(l-1)}$ is the output from previous layer, and $\mathbf{h}_i^{(l=0)}$ is the input feature.

Neighbors contribute information differently, e.g., some hyperlinked news articles report similar events, while others are coincidence. Thus, we design attention to distinguish i's neighbors.

$$a_{ij} = \text{softmax}\Big(\text{sigmoid}(\mathbf{b}^{(l)\top}[\tilde{\mathbf{h}}_i^{(l)}||\tilde{\mathbf{h}}_j^{(l)}])\Big) \quad \text{where} \quad j \in \mathcal{N}(i). \quad (5)$$

$\mathcal{N}(i)$ is the neighbor set, $[\cdot||\cdot]$ is concatenation, $\mathbf{b}^{(l)}$ is learnable parameter. Finally, we propagate topics of i's neighbors to i by

$$\mathbf{h}_i^{(l)} = f_{\text{act}}\Big(\frac{1}{2}(\tilde{\mathbf{h}}_i^{(l)} + \sum_{j\in\mathcal{N}(i)} a_{ij}\tilde{\mathbf{h}}_j^{(l)})\Big). \quad (6)$$

$f_{\text{act}}(\cdot)$ is activation. We set it to identity function $f_{\text{act}}(x) = x$ only for the final convolutional layer and $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ for other layers. We obtain $\mathbf{h}_i^{(l)}$ as the output from the l-th layer, containing both text content and i's neighborhood. We repeat convolutional process at Eq. 4–6 for maximum L layers, and obtain $\mathbf{h}_i = \mathbf{h}_i^{(L)} \in \mathbb{R}^K$ as the output from the encoder. K is the number of topics.

We aim to use $\mathbf{h}_i$ as concentration parameter of Dirichlet distribution to draw document i's topic distribution, i.e., $\mathbf{z}_i \sim \text{Dir}(\mathbf{h}_i)$. However, Dirichlet's concentration parameter should be positive, while $\mathbf{h}_i$ from Eq. 6 is obtained by identity function with both positive and negative values. To solve this problem, we have

$$\boldsymbol{\alpha}_i = \max(10^{-12}, \text{softplus}(\mathbf{h}_i)). \quad (7)$$

$\text{softplus}(x) = \log(1+\exp(x))$ outputs positive values. Threshold 10^{-12} avoids extremely small values. $\boldsymbol{\alpha}_i$ now is a positive vector. In the next subsection, we will consider $\boldsymbol{\alpha}_i$ as concentration parameter to draw i's topic distribution, $\mathbf{z}_i \sim \text{Dir}(\boldsymbol{\alpha}_i)$.

4.1 Dirichlet Reparameterization

Since Dirichlet prior in LDA [2] has shown promise in improving topic quality, we are motivated to use Dirichlet as a prior for topic distribution to enhance topic quality, $\mathbf{z}_i \sim \text{Dir}(\boldsymbol{\alpha}_i)$. However, Dirichlet is not location scale family, thus it is difficult to directly sample topic distributions from it, which hinders reparameterization. To solve this problem, we use rejection sampling [28]. Laplace approximation [19] is an alternative method to approximate Dirichlet. In this paper, we use rejection sampling due to two reasons. For one, there exits an effective proposal function for Gamma distribution, which can be used to simulate Dirichlet. For the other, empirically rejection sampling has shown better approximation precision than Laplace [20].

Dirichlet $\text{Dir}(\boldsymbol{\alpha}_i)$ can be simulated by Gamma distributed random variables. If $z_{i,k} \sim \Gamma(\alpha_{i,k})$ where $\Gamma(\cdot)$ is Gamma, then

$$\mathbf{z}_i = \left[\frac{z_{i,k}}{\sum_{k'=1}^K z_{i,k'}}, ..., \frac{z_{i,K}}{\sum_{k'=1}^K z_{i,k'}}\right] \sim \text{Dir}(\boldsymbol{\alpha}_i). \quad (8)$$

As long as we can sample topics from Gamma $\Gamma(\alpha_{i,k})$, we can approximate Dirichlet by the normalization at Eq. 8. Now the problem is how to sample topics from Gamma, i.e., $z_{i,k} \sim \Gamma(\alpha_{i,k})$.

Fortunately, there exits a proposal function for Gamma distribution $\Gamma(\alpha_{i,k})$ below [20], such that $f(\alpha_{i,k}) \geq \Gamma(\alpha_{i,k})$.

$$z_{i,k} = f(\alpha_{i,k}) = (\alpha_{i,k} - \frac{1}{3})(1 + \frac{\epsilon}{\sqrt{9\alpha_{i,k} - 3}})^3, \quad \epsilon \sim \mathcal{N}(0,1). \quad (9)$$

Sampling from proposal function $f(\alpha_{i,k})$ is possible by first sampling a variable ϵ from a Gaussian $\mathcal{N}(0,1)$, then putting into Eq.

9 to obtain the sampled topics $z_{i,k}$. However, sampled topics $z_{i,k}$ from this proposal function do not strictly obey the target Gamma distribution, since some samples should be rejected. Lower the concentration parameter $\alpha_{i,k}$, lower the acceptance rate [20]. If the acceptance rate is overly low, we need multiple repetitions of rejection sampling to obtain one valid sample. To this end, we aim to increase the acceptance rate by increasing concentration parameter $\alpha_{i,k}$. Fortunately, [38] suggests the following solution: since we seek to sample topics $z_{i,k}$ from Gamma distribution by $z_{i,k} \sim \Gamma(\alpha_{i,k})$, we can equivalently rewrite this sampling by

$$z_{i,k} = \bar{z}_{i,k} \prod_{c=1}^{C} u_c^{\frac{1}{\alpha_{i,k}+c-1}} \quad \text{where} \quad \bar{z}_{i,k} \sim \Gamma(\alpha_{i,k} + C). \tag{10}$$

Positive integer C is a hyperparameter to boost concentration parameter $\alpha_{i,k}$, so that sampling process $\bar{z}_{i,k} \sim \Gamma(\alpha_{i,k} + C)$ has a higher concentration parameter $\alpha_{i,k} + C$ now, which leads to a higher acceptance rate by rejection sampling at Eq. 9. As in [38], $C = 10$ provides an acceptance rate higher than 0.99, thus we set $C = 10$ and accept all samples for simplicity. After obtaining $\bar{z}_{i,k}$ as a valid sample at Eq. 9, we use Eq. 10 to calculate $z_{i,k}$. Above, we use one topic $z_{i,k}$ for illustration. We repeat this process for K (the number of topics) times, and obtain K valid sampled topics $\{z_{i,k}\}_{k=1}^{K}$ from Gamma. Finally, we use Eq. 8 to normalize them and obtain K-dimensional topic distribution $\mathbf{z}_i$. To summarize, $\mathbf{z}_i \sim \text{Dir}(\boldsymbol{\alpha}_i) = q(\mathbf{z}_i)$. Variational posterior $q(\mathbf{z}_i)$ contains both graph convolutional encoder and Dirichlet reparameterization.

KL Divergence. We now turn to the formulation of KL divergence between variational posterior $q(\mathbf{z}_i)$ and the predefined Dirichlet prior $p(\mathbf{z}) = \text{Dir}(\boldsymbol{\alpha}^0)$, which has an analytical form.

$$\begin{aligned}\text{KL}[q(\mathbf{z}_i)||p(\mathbf{z})] &= \log\Gamma(\sum_{k=1}^{K}\alpha_{i,k}) - \log\Gamma(\sum_{k=1}^{K}\alpha_k^0) + \sum_{k=1}^{K}\log\Gamma(\alpha_k^0) \\ &- \sum_{k=1}^{K}\log\Gamma(\alpha_{i,k}) + \sum_{k=1}^{K}(\alpha_{i,k} - \alpha_k^0)(\Psi(\alpha_{i,k}) - \Psi(\sum_{k=1}^{K}\alpha_{i,k}))\end{aligned} \tag{11}$$

where $\Psi(\cdot)$ is digamma function [20].

4.2 Barycentric Decoding (DBN)

Since a document i is usually represented by two distributions, latent topics $\mathbf{z}_i$ and observed words $\mathbf{d}_i$, they consistently reflect the same document. We thus aim to push topic distribution $\mathbf{z}_i$ to word distribution $\mathbf{d}_i$. Though their dimensions are different, $K \neq |\mathcal{V}|$, OT solves the problem. We seek to optimize $\min d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_i)$.

Traditional models use topic distribution to generate its own content. We discover that two linked documents likely share similar topics, though their texts are different. Motivated by this intuition, we define *barycentric topic modeling* using optimal transport to push topics of one document to its neighbors.

$$\min \sum_{j \in \mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j). \tag{12}$$

a_{ij} is defined at Eq. 5. By moving topic $\mathbf{z}_i$ to i's neighbors $\mathcal{N}(i)$, $\mathbf{z}_i$ becomes the barycenter of $\mathcal{N}(i)$. Network structure is captured.

We now define the cost matrix at Eq. 12. Each element c_{kw} in $\mathbf{C} \in \mathbb{R}^{K \times |\mathcal{V}|}$ specifies the dissimilarity between topic k and word w. We set $c_{kw} = 1 - \cos(\mathbf{t}_k, \mathbf{e}_w)$ where $\mathbf{t}_k$ is the randomly initialized topic embedding, and $\mathbf{e}_w$ is pre-trained word embedding [39], [40]. External knowledge is naturally incorporated by cost matrix, and helps alleviate word sparsity problem of short documents.

Decoding. Eq. 12 pushes topic distribution $\mathbf{z}_i$ towards the word distribution of i's neighbors, which is similar to a decoder. We explicitly design another decoder $\hat{\mathbf{d}}_i = \phi(\mathbf{z}_i) = \text{softmax}((2 - \mathbf{C})^\top \mathbf{z}_i)$ to generate the content of *neighboring documents*. $2 - \mathbf{C}$ is decoding parameter, since cost matrix $\mathbf{C}$ captures topic-word dissimilarity. In topic model literature, $2 - \mathbf{C}$ is also called topic-word distribution mentioned in Def. 3.1. $\mathbf{C}$ is defined by $c_{kw} = 1 - \cos(\mathbf{t}_k, \mathbf{e}_w)$, where cosine similarity has range $[-1, 1]$, thus c_{kw} has range $[0, 2]$. As in [2], [18], decoding parameter or topic-word distribution is usually a positive matrix where each element is the probability of a word belonging to a certain topic. We define $2 - \mathbf{C}$ to make our decoding parameter positive.

The log-likelihood is $\sum_{j \in \mathcal{N}(i)} a_{ij} \mathbf{d}_j^\top \log \hat{\mathbf{d}}_i = \sum_{j \in \mathcal{N}(i)} a_{ij} l(\mathbf{z}_i, \mathbf{d}_j)$. Weights a_{ij} are Eq. 5. We generate content of neighbors in a 1-to-N process. Loss function is

$$\begin{aligned}\mathcal{J}_{\text{DBN}} = \sum_{j \in \mathcal{N}(i)} &\mathbb{E}_{\mathbf{z}_i \sim q(\mathbf{z}_i)}[a_{ij}(-l(\mathbf{z}_i, \mathbf{d}_j) + \lambda_{\text{OT}} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j))] \\ &+ \lambda_{\text{KL}} \text{KL}[q(\mathbf{z}_i)||p(\mathbf{z})].\end{aligned} \tag{13}$$

$q(\mathbf{z}_i)$ contains both graph convolutional encoder and Dirichlet reparameterization. Hyperparameter λ_{OT} balances log-likelihood and optimal transport barycenter, and λ_{KL} controls KL divergence.

Compared to topic models with OT, e.g., NSTM [6], we point out two main extensions. First, NSTM models each document individually and does not have document network, while we design a graph convolutional encoder to model both text and network connectivity. Second, motivated by the success of Dirichlet prior in LDA [2], we design a Dirichlet optimal transport prior to improve topic quality. NSTM does not impose any prior, likely suffering over-fitting and worse topic interpretability.

Compared to VGAE, whose embeddings do not enjoy semantic interpretability, the decoding term of our model at Eq. 13 contains OT barycenter, which incorporates pre-trained word embeddings to define cost matrix $\mathbf{C}$. Decoding parameter $2 - \mathbf{C} \in \mathbb{R}^{K \times |\mathcal{V}|}$ is topic-word distribution. Each row is the distribution of a topic over the vocabulary, and the keywords of that topic are the highest values on that row. Thus the learned topic distributions are semantically interpretable by topic-word distribution.

Theorem 4.1. *Given $\boldsymbol{C}$, $|\mathcal{V}| \geq 8$, and a_{ij}, let $\boldsymbol{z}_j = \arg\min_{\boldsymbol{z}_j} d_{\boldsymbol{C}}(\boldsymbol{z}_j, \boldsymbol{d}_j)$ be the best solution of its individual optimal transport without barycenter, $\boldsymbol{z}_i = \arg\min_{\boldsymbol{z}_i} \sum_{j \in \mathcal{N}(i)} a_{ij} d_{\boldsymbol{C}}(\boldsymbol{z}_i, \boldsymbol{d}_j)$ be the best solution of optimal transport barycenter Eq. 12, we have*

$$\begin{aligned}-\sum_{j \in \mathcal{N}(i)} a_{ij} l(\boldsymbol{z}_i, \boldsymbol{d}_j) &\geq \sum_{j \in \mathcal{N}(i)} a_{ij} d_{\boldsymbol{C}}(\boldsymbol{z}_i, \boldsymbol{d}_j) \\ &\geq \sum_{j \in \mathcal{N}(i)} a_{ij} d_{\boldsymbol{C}}(\boldsymbol{z}_j, \boldsymbol{d}_j).\end{aligned} \tag{14}$$

The first inequality is similar to [6], OT barycentric distance $\sum_{j \in \mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j)$ is the lower bound of the negative log-likelihood $-\sum_{j \in \mathcal{N}(i)} a_{ij} l(\mathbf{z}_i, \mathbf{d}_j)$. Thus, OT barycentric distance at Eq. 13 is a regularizer to avoid negative log-likelihood becoming overly low and resulting in over-fitting. Here, the constraint of vocabulary size, $|\mathcal{V}| > 8$, is required at one step in the proof to make sure the inequality is satisfied (Eq. 22). Almost all the real-word corpora have hundreds or thousands of words in vocabulary, thus $|\mathcal{V}| > 8$ is not a strict constraint, i.e., the first inequality is satisfied most of the time. The second inequality

(a) Geometric interpretation of DBN. Document D contains quite different content from A.

(b) Geometric interpretation of D²BN. Document A shares common neighbors with D, and its topic barycenter is pushed to D.

Fig. 1: Geometric interpretation of DBN and D^{2}BN.

Algorithm 1 Learning Algorithm of DBN and D^{2}BN

Input: Document network $\mathcal{G} = (\mathcal{D}, \mathcal{E})$, pre-trained word embeddings $\{\mathbf{e}_w\}_{w\in\mathcal{V}}$, and structure embeddings $\{\mathbf{x}_i\}_{i=1}^N$ for D^{2}BN, number of topics K, λ_{OT}, λ_{KL}, λ_s, γ.

Output: Encoder $q(\cdot)$, topic embeddings $\{\mathbf{t}_k\}_{k=1}^K$.

Initialize all parameters.
while not converged **do**
  Sample a batch of documents $\{\mathbf{d}_b\}_{b=1}^B$.
  Encode $\{\mathbf{d}_b\}_{b=1}^B$ and neighbors $\{\mathcal{N}(b)\}_{b=1}^B$ by $q(\cdot)$.
  Generate neighboring content by $\phi(\cdot)$, and neighboring adjacency vector by $\sigma(\cdot)$ for D^{2}BN.
  Obtain optimal transport plan $\mathbf{T}_{ij}^*$ for DBN, and $\mathbf{T}_{ij}'^*$ for D^{2}BN, by solving OT $d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j)$ and $d_{\mathbf{C}'}(\mathbf{z}_i, \boldsymbol{\varepsilon}_j)$ using Sinkhorn Iteration at Algo. 2, for each document i and each of its neighbors $j \in \mathcal{N}(i)$.
  Loss function Eq. 13 for DBN, Eq. 17 for D^{2}BN.
  Update parameters by Adam optimizer.
end while

Algorithm 2 Sinkhorn Iteration of DBN and D^{2}BN

Input: Document i's topic distribution $\mathbf{z}_i$, neighbor j's word distribution $\mathbf{d}_j$ where $j \in \mathcal{N}(i)$, neighbor j's adjacency vector $\boldsymbol{\varepsilon}_j$ for D^{2}BN, cost matrix $\mathbf{C}$ for DBN and $\mathbf{C}'$ for D^{2}BN, γ.

Output: OT plan $\mathbf{T}^*$ and OT distance $d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j)$ for DBN, OT plan $\mathbf{T}'^*$ and OT distance $d_{\mathbf{C}'}(\mathbf{z}_i, \boldsymbol{\varepsilon}_j)$ for D^{2}BN

Initialize $\boldsymbol{\pi} = \frac{\mathbf{1}_K}{K}$ and $\boldsymbol{\Phi} = \exp(-\frac{\mathbf{C}}{\gamma})$ for DBN, $\boldsymbol{\pi}' = \frac{\mathbf{1}_K}{K}$ and $\boldsymbol{\Phi}' = \exp(-\frac{\mathbf{C}'}{\gamma})$ for D^{2}BN.
while not converged **do**
  $\boldsymbol{\beta} = \dfrac{\mathbf{d}_j}{\boldsymbol{\Phi}^\top \boldsymbol{\pi}}$ for BDN, $\boldsymbol{\beta}' = \dfrac{\boldsymbol{\varepsilon}_j}{\boldsymbol{\Phi}'^\top \boldsymbol{\pi}'}$ for D^{2}BN
  $\boldsymbol{\pi} = \dfrac{\mathbf{z}_i}{\boldsymbol{\Phi}\boldsymbol{\beta}}$ for DBN, $\boldsymbol{\pi}' = \dfrac{\mathbf{z}_i}{\boldsymbol{\Phi}'\boldsymbol{\beta}'}$ for D^{2}BN
end while
Obtain OT plan $\mathbf{T}^* = \text{diag}(\boldsymbol{\pi})\boldsymbol{\Phi}\text{diag}(\boldsymbol{\beta})$ for DBN, $\mathbf{T}'^* = \text{diag}(\boldsymbol{\pi}')\boldsymbol{\Phi}'\text{diag}(\boldsymbol{\beta}')$ for D^{2}BN
Obtain OT distance $d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) =< \mathbf{T}^*, \mathbf{C} >$ for DBN, $d_{\mathbf{C}'}(\mathbf{z}_i, \boldsymbol{\varepsilon}_j) =< \mathbf{T}'^*, \mathbf{C}' >$ for D^{2}BN

reveals that 1-to-N content generation by optimizing barycenter $\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j)$ is the upper bound of traditional 1-to-1 content generation without network structure, and provides a tighter regularizer. See Appendix A for complete proofs.

4.3 Double Barycentric Decoding (D^{2}BN)

DBN implicitly captures network by generating neighbor content. We design an extended model D^{2}BN, for **D**irichlet Optimal Transport **D**ouble **B**arycenter for Document **N**etworks, which explicitly models network by inducing a double OT barycenter.

We encode a document i into K-dimensional topic distribution by $\mathbf{z}_i \sim q(\mathbf{z}_i)$. Intuitively, two documents sharing common neighbors likely share similar topics, even when not directly connected, e.g., two news articles with common hyperlinked related articles tend to report similar events; papers with common citations likely discuss similar research. Adjacency matrix $\mathcal{E}$ preserves neighborhood information. Each row is the connectivity of a document. We represent the neighbor distribution of document i by $\boldsymbol{\varepsilon}_i$ with each element $\varepsilon_{ij} = \frac{e_{ij}}{|\mathcal{N}(i)|}$. $e_{ij} = 1$ if there is a link between i and j, $e_{ij} = 0$ otherwise. We also design an OT barycentric modeling between topic and structure spaces for 1-to-N generation.

$$\sum_{j\in\mathcal{N}(i)} \mathbb{E}_{\mathbf{z}_i\sim q(\mathbf{z}_i)}[a_{ij}(-l(\mathbf{z}_i, \boldsymbol{\varepsilon}_j) + \lambda_{\text{OT}} d_{\mathbf{C}'}(\mathbf{z}_i, \boldsymbol{\varepsilon}_j))]. \tag{15}$$

Besides text content, we allow i's topics $\mathbf{z}_i$ to also generate the adjacency vector $\boldsymbol{\varepsilon}_j$ of its neighbors. Two similar adjacency vectors of two corresponding documents force their topics to be similar. Topics are thus enhanced by network structure explicitly.

We first define cost matrix $\mathbf{C}' \in \mathbb{R}^{K\times N}$ of $d_{\mathbf{C}'}(\mathbf{z}_i, \boldsymbol{\varepsilon}_j)$ at Eq. 15. Each element c'_{ki} preserves structural information between topic k and document i. $N = |\mathcal{D}|$ is the total number of documents. To keep consistent with $\mathbf{C}$ above, we similarly define $c'_{ki} = 1 - \cos(\mathbf{t}_k, \mathbf{s}_i)$. $\mathbf{t}_k$ is the same topic embedding as above, and $\mathbf{s}_i$ is the structure embedding of document i.

$$\mathbf{s}_i = f(\mathbf{W}_s \mathbf{x}_i + \mathbf{b}_s). \tag{16}$$

Consistently with pre-trained word embeddings, $\mathbf{x}_i$ is pre-trained structure embedding. Different from $\mathbf{x}_i$, $\mathbf{s}_i$ is the projected structure embedding and is derived by one-layer neural network. We feed $\mathbf{x}_i$ by pre-trained structure embeddings, e.g., DeepWalk [41].

TABLE 2: Dataset statistics.

Name	#Documents	#Links	Vocabulary	#Labels	#Words/doc
DS	1,703	3,234	3,134	9	58.0
ML	3,087	8,573	3,040	7	64.2
PL	2,597	7,754	3,106	9	64.0
Aminer	42,564	40,269	4,094	10	6.5
Web	445,657	565,502	10,015	N.A.	79.8

$\mathbf{W}_s$ and $\mathbf{b}_s$ are parameters that project $\mathbf{x}_i$ to the same topic embedding space. The structure decoder is $\hat{\boldsymbol{\varepsilon}}_i = \sigma(\mathbf{z}_i) = \text{softmax}((2-\mathbf{C}')^\top \mathbf{z}_i)$, log-likelihood at Eq. 15 is $l(\tilde{\mathbf{z}}_i, \boldsymbol{\varepsilon}_j) = \boldsymbol{\varepsilon}_j^\top \log \hat{\boldsymbol{\varepsilon}}_i$.

Finally, integrating both content and structure modeling, we have loss function for double barycentric topic modeling D^{2}BN.

$$\mathcal{J}_{\text{D}^2\text{BN}} = \sum_{j\in\mathcal{N}(i)} \mathbb{E}_{\mathbf{z}_i\sim q(\mathbf{z}_i)}\Big[a_{ij}\Big(-l(\mathbf{z}_i,\mathbf{d}_j) + \lambda_{\text{OT}} d_{\mathbf{C}}(\mathbf{z}_i,\mathbf{d}_j) + \lambda_s(-l(\mathbf{z}_i,\boldsymbol{\varepsilon}_j) + \lambda_{\text{OT}} d_{\mathbf{C}'}(\mathbf{z}_i,\boldsymbol{\varepsilon}_j))\Big)\Big] + \lambda_{\text{KL}}\text{KL}[q(\mathbf{z}_i)||p(\mathbf{z})]. \tag{17}$$

λ_s controls content and structure modeling. This joint decoding pushes topics $\mathbf{z}_i$ to be the *double OT barycenter* of both content and structure space. DBN is a special case of D^{2}BN if $\lambda_s = 0$.

4.4 Optimization and Analysis

Geometric interpretation. For DBN at Fig. 1(a), the black dot is document A's topic distribution, which is also the topic barycenter of A and its neighbors B and C. We minimize weighted OT distance between $\mathbf{z}_A$ and not only $\mathbf{d}_A$, but also $\mathbf{d}_B$ and $\mathbf{d}_C$. For D^{2}BN at Fig. 1(b), structure space contains adjacency vectors, which regularize topic distributions. Although D contains different content from A in semantic space, they share common neighbors in structure space. A's topic distribution is also pushed to D. Such a double barycenter enjoys both semantic and structure information.

Optimization. We summarize the learning process in Algo. 1. Here we apply minibatch optimization. Line 6 calculates optimal transport plan $\mathbf{T}^*$ and $\mathbf{T}'^*$ by Sinkhorn iteration [4] at Algo. 2. After training convergence, we infer the topic distribution of a previously unseen document $\mathbf{d}'$ simply by the encoder $\mathbf{z}' \sim q(\mathbf{z}')$.

Complexity. To better understand our models, we provide complexity. Encoding has $\mathcal{O}(\deg_{\max}^L(WK + \deg_{\max} K))$. $\deg_{\max}$ is the maximum number of neighbors, and W is the dimension of word embeddings. Dirichlet reparameterization and KL divergence has $\mathcal{O}(CK + \alpha^0 K + K^2)$. Here, $\boldsymbol{\alpha}^0 = [\alpha_1^0, ..., \alpha_K^0]$ is the concentration parameter of Dirichlet prior where $\alpha_1^0 = ... = \alpha_K^0 = \alpha^0$. Decoder is $\mathcal{O}(K|\mathcal{V}|)$ for DBN and $\mathcal{O}(K(|\mathcal{V}| + N))$ for D^{2}BN. OT optimization is $\mathcal{O}(WK|\mathcal{V}|)$ for DBN and $\mathcal{O}(WK(|\mathcal{V}|+N))$ for D^{2}BN. Putting all components together and removing trivial terms, we have $\mathcal{O}(\deg_{\max}^L(WK + \deg_{\max} K) + K^2 + WK|\mathcal{V}|)$ for DBN and $\mathcal{O}(\deg_{\max}^L(WK+\deg_{\max} K)+K^2+WK(|\mathcal{V}|+N))$ for D^{2}BN.

Short comment on running time. Our focus in this paper is model effectiveness, not running efficiency. But for completeness, we still briefly report running time. On the largest dataset Web, DBN took 30 min to converge, and D^{2}BN took 90 min, since double barycenter brings additional complexity. All the experiments were done on a Tesla K80 GPU with 11441MiB. We consider speeding up the training with *online learning* as future work.

5 Experiments

The goal of experiments is to evaluate the quality of learned topics by our models on evaluation tasks, including document classification, clustering, link prediction, topic analysis, etc.

Datasets. Cora [42] is a corpus of academic papers with abstract as content and citations as links. We create three independent datasets, Data Structure (**DS**), Machine Learning (**ML**), and Programming Language (**PL**). **Aminer** [43] is another citation network with titles as the only content. We further create a **Web** page hyperlink network [44]. Each page is a news article. Hyperlinks point to relevant pages. Table 2 shows statistics.

Baselines. We consider three categories of baselines models.

1) **Topic models without network structure**, ProdLDA [19], Dirichlet VAE [20], ETM [24], WLDA [26], NSTM [6], and GTM [45]. ProdLDA and DVAE use Dirichlet as prior. ETM is a neural model with pre-trained word embeddings, WLDA applies Wasserstein distance, and NSTM uses optimal transport. GTM is designed with graph neural networks. They do not incorporate network structure. By comparison, we show the advantage of jointly modeling text and network connectivity.
2) **Topic models for document networks**, RTM [3], NRTM [27], Adjacent-Encoder [29], LANTM [31], GTNN [30], GRTM [46], GNCTM [47], NetDTM [48], VGATM [49]. NRTM and Adjacent-Encoder are built on Auto-Encoders. LANTM, GTNN, GRTM, GNCTM, VGATM are designed with graph neural networks. They consider both text and links, but no one models pre-trained word embeddings. The comparison to them validates the effect of pre-trained word embeddings. LANTM extends VGAE with Gaussian prior. The comparison to LANTM shows the utility of Dirichlet prior. NetDTM is designed with pre-trained word embeddings, but not OT barycenter.
3) **Graph embedding.** There are models learning node embeddings on attributed graphs in an unsupervised method. Strictly speaking, they are not topic models, nor baselines. For completeness, we still compare to VGAE [1] and PGCL [50].

Hyperparameters are set based on validation set (see below classification on how we split validation set). For ProdLDA, DVAE, and our models, we set 1 as concentration parameter for Dirichlet prior. For models with word embeddings, including ours, we use 300D GloVe. Structure embeddings $\mathbf{x}_i$ are pre-trained by DeepWalk [41]. We set $L = 3$ convolutional layers. Dropout rate is 0.6. $\lambda_{\text{OT}} = 2$ and $\lambda_s = 1$ after searching in $[0.5, 1, 2, 4, 10]$. $\lambda_{\text{KL}} = 0.001$ for KL divergence. $\gamma = 20$ for OT. Each result is obtained by 5 independent runs. We report both mean and std.dev.

5.1 Quantitative Evaluation

5.1.1 Document Classification

Documents within the same category share similar topics. As in LDA [2], we do document classification. We randomly split 80% documents for training (among which 10% are for validation), 20% for testing. During training, we observe training documents and links within them. Labels are never involved when training. After convergence, we infer topics of test documents and classify them with κNN [28]. We input topic distributions of training documents to κNN and predict the labels of test documents.

We first vary the number of topics K and report micro f1 score with 5NN at Fig. 2(a-d). LANTM cannot run on large dataset Aminer even on a machine with 256GB, thus is excluded. Overall, document network models generally perform better than models with content only, since network structure indicates topic

(a) Number of topics on DS (b) Number of topics on ML (c) Number of topics on PL (d) Number of topics on Aminer

(e) Number of neighbors on DS (f) Number of neighbors on ML (g) Number of neighbors on PL (h) Number of neighbors on Aminer

Fig. 2: Document classification with Micro F1 score when varying number of topics (a-d) and number of nearest neighbors κ for κNN (e-h).

TABLE 3: Document classification with Micro F1 (left) and Macro F1 (right) at $K = 64$. Results are in percentage. LANTM cannot run on Aminer even on a machine with 256GB. Web dataset does not have ground-truth labels, thus can not evaluate document classification.

Category	Model	Micro F1 score				Macro F1 score			
		DS	ML	PL	Aminer	DS	ML	PL	Aminer
Topic models w/o network structure	ProdLDA	51.4±1.1	67.0±0.6	51.8±0.6	40.5±0.0	40.1±4.3	67.1±0.7	48.5±1.2	14.6±0.9
	DVAE	54.7±2.2	68.9±0.6	55.7±1.3	66.1±0.5	45.8±1.2	65.5±1.4	49.3±1.5	44.6±1.6
	ETM	42.2±2.4	53.9±1.8	45.0±2.1	53.2±0.7	31.1±3.4	50.4±1.7	40.3±1.3	25.7±0.8
	WLDA	34.0±3.6	32.4±1.1	30.5±3.1	47.8±2.1	24.7±3.0	30.5±1.0	24.3±3.8	18.6±1.9
	NSTM	42.5±1.8	46.3±1.8	42.0±1.6	48.4±0.3	34.9±3.3	42.9±1.8	34.6±0.7	20.0±0.6
	GTM	39.1±1.6	45.9±1.4	40.8±1.6	65.2±0.2	29.6±1.3	42.1±1.7	34.4±2.0	43.3±0.7
Topic models for document networks (LANTM cannot run on large dataset Aminer)	RTM	50.1±2.3	63.7±0.9	52.3±0.7	53.0±0.7	41.3±2.2	58.6±1.6	44.4±0.7	25.1±0.7
	NRTM	38.2±1.2	45.5±1.2	39.2±1.5	47.7±3.9	32.3±1.6	41.7±1.4	34.1±0.9	13.5±3.2
	Adjacent-Encoder	58.8±1.2	72.8±0.6	60.0±1.7	59.5±0.2	54.6±1.5	72.8±0.8	55.3±1.6	46.9±0.5
	LANTM	56.8±2.4	71.8±1.0	62.6±1.3	N.A.	54.7±0.8	70.0±1.3	55.6±1.9	N.A.
	GTNN	52.9±1.4	68.6±1.1	59.5±2.3	65.5±0.4	42.8±3.3	67.8±1.1	59.4±1.5	41.8±0.5
	GRTM	56.5±1.9	51.7±2.3	45.2±1.0	70.7±0.2	50.2±2.0	48.3±1.9	37.5±2.1	48.8±0.5
	GNCTM	63.7±1.4	77.8±1.1	65.5±3.8	69.3±1.2	59.6±2.0	75.9±2.8	60.1±4.0	47.6±2.6
	NetDTM	62.3±1.0	74.8±1.0	63.0±1.1	68.1±0.1	58.0±1.4	73.1±1.1	57.2±1.2	46.6±0.7
	VGATM	63.6±2.1	74.8±1.3	64.3±0.9	57.2±0.7	58.8±2.2	73.0±1.3	57.9±1.2	42.2±0.6
Graph embedding	VGAE	39.8±2.0	56.6±1.7	47.6±3.4	64.7±0.5	40.3±3.5	61.5±2.3	50.0±1.5	45.0±1.4
	PGCL	62.9±1.2	74.9±1.2	64.7±1.1	69.7±0.4	57.5±2.9	72.9±1.6	56.3±0.7	48.9±0.8
Our proposed models	DBN	**66.2±1.4**	78.4±0.8	67.3±0.5	71.2±0.4	**62.7±1.6**	77.3±0.6	61.2±0.4	50.9±1.0
	D²BN	65.8±1.6	**81.1±1.2**	**71.3±0.7**	**72.3±0.3**	62.5±2.0	**79.8±1.3**	**64.7±1.1**	**51.8±0.2**

similarity among documents, and modeling it can bring similar documents closer, thus achieving better results. The improvement of our models over network baselines, especially on Aminer with short texts, verifies the advantage of pre-trained word embeddings to alleviate word sparsity. Both our models perform stably across different number of topics. D^{2}BN generally achieves better results than DBN, since double barycenter regularizes topic distributions by explicitly modeling network. Since most models begin to plateau at 64 topics, we keep $K = 64$ for subsequent experiments.

We then vary the number of nearest neighbors κ for κNN classifier. Fig. 2(e-h) shows the results at $K = 64$ topics. Overall, both our models outperform baselines across different number of nearest neighbors. Most models present a stable performance. GNCTM jointly models text and network structure, and is the best-performing baseline. It is competitive with DBN on ML dataset, but is still worse than D^{2}BN. LANTM extends VGAE as a topic model with Gaussian prior. The comparison to LANTM verifies the advantage of Dirichlet prior. As shown by previous works [20], Dirichlet forces topics to be sparser than Gaussian and achieves lower reconstruction error, thus improving classification. Table 3 summarizes both micro and macro f1 scores at $K = 64$.

5.1.2 Document Clustering

As in [29], we evaluate topic quality by document clustering. After training, we use K-Means [28] to cluster test documents. Labels are involved only for evaluating the clustering quality by Normalized Mutual Information (NMI) [51]. Table 4(left) shows the results. Benefiting from both text and network structure, Adjacent-Encoder presents the best results among baselines. Though DBN is competitive with Adjacent-Encoder, the advanced D^{2}BN still significantly outperforms the latter, since double barycenter enhances topic quality by both semantic and structure generation. The comparison to NRTM, LANTM, and VGAE verifies the advantage of Dirichlet prior, which improves topic quality by forcing topic distributions to be sparser than Gaussian.

5.1.3 Link Prediction

A good model should derive similar topics for potentially linked documents. As in RTM [3], given two documents, we use their

TABLE 4: Document clustering NMI (left) and Link prediction AUC (right) at $K = 64$. Results are in percentage. LANTM cannot run on Aminer and Web even on a machine with 256GB. Web dataset does not have ground-truth labels, thus can not evaluate document clustering.

Category	Model	Document Clustering NMI				Link Prediction AUC				
		DS	ML	PL	Aminer	DS	ML	PL	Aminer	Web
Topic models w/o network structure	ProdLDA	29.9±2.2	38.4±1.2	26.4±2.0	8.7±0.8	76.8±0.5	80.7±0.6	75.3±0.3	64.2±0.9	82.4±0.0
	DVAE	28.2±1.1	35.0±0.3	24.8±0.5	25.8±0.4	74.0±1.1	77.6±0.5	75.0±0.3	86.8±0.1	88.3±0.0
	ETM	19.3±1.7	21.5±2.1	19.8±1.1	10.4±1.9	71.0±1.3	68.7±1.8	69.3±0.6	68.8±0.7	72.3±0.2
	WLDA	11.0±0.0	8.7±0.0	9.2±0.0	9.9±0.0	60.3±0.5	55.0±2.5	57.1±1.0	70.4±2.2	79.3±0.5
	NSTM	8.0±0.7	7.8±1.3	9.1±0.8	3.0±0.7	62.1±1.4	63.6±1.3	63.7±0.6	62.4±0.4	67.0±0.8
	GTM	16.8±2.3	17.8±1.7	15.8±0.8	19.5±0.5	63.8±1.5	61.5±2.1	65.9±2.1	77.1±0.3	80.8±0.1
Topic models for document networks (LANTM cannot run on large dataset Aminer and Web)	RTM	6.3±3.2	14.7±2.4	9.3±2.2	8.7±0.8	70.6±0.6	71.6±1.9	69.4±0.5	74.4±0.5	78.4±0.1
	NRTM	17.6±0.9	12.7±0.7	15.2±0.9	15.0±1.5	71.0±2.1	64.8±0.9	67.6±0.7	66.3±1.3	62.1±0.7
	Adjacent-Encoder	31.8±0.9	43.6±1.0	28.4±1.2	27.6±0.2	81.7±0.4	84.7±0.2	83.2±0.1	88.3±0.1	73.2±0.0
	LANTM	24.0±1.6	19.7±2.7	20.7±1.4	N.A.	78.4±0.6	78.7±0.9	78.7±1.2	N.A.	N.A.
	GTNN	19.5±2.1	28.6±1.8	22.1±1.2	21.0±0.9	71.5±1.1	74.5±0.4	72.4±0.1	82.0±0.6	74.3±0.2
	GRTM	30.4±3.3	22.5±1.4	18.6±1.7	27.4±0.3	79.3±0.5	71.7±0.4	67.7±0.8	86.9±0.2	85.4±1.4
	GNCTM	28.0±3.4	32.5±1.6	21.4±3.8	18.5±1.4	86.2±1.1	84.5±1.9	86.0±2.2	85.5±1.1	87.8±0.4
	NetDTM	29.7±0.8	33.3±1.2	24.9±1.0	23.2±0.4	84.2±0.7	81.1±0.6	82.2±0.3	82.9±0.2	87.5±0.0
	VGATM	28.7±1.1	36.5±1.5	26.9±1.2	17.4±0.4	87.8±0.6	85.1±0.7	87.0±0.9	74.6±0.3	82.5±0.0
Graph embedding	VGAE	16.9±1.6	21.7±0.5	18.0±1.5	18.6±1.4	63.4±2.0	64.8±2.0	65.3±0.7	78.3±1.1	87.4±0.2
	PGCL	29.6±3.2	30.4±2.3	28.6±1.8	16.8±1.5	89.2±0.9	87.2±0.9	88.8±0.5	86.7±0.5	83.0±0.3
Our proposed models	DBN	33.0±0.9	33.8±0.4	28.0±0.6	22.7±0.2	89.6±0.5	86.2±0.9	88.1±0.4	89.2±0.2	**89.7±0.0**
	D^{2}BN	**38.9±1.7**	**45.3±0.4**	**37.9±1.0**	**28.9±1.0**	**90.1±0.7**	**91.4±0.4**	**92.7±0.3**	**92.0±0.1**	88.2±0.1

TABLE 5: Topic coherence NPMI (left, in percentage) and perplexity (right, lower is better) at $K = 64$. LANTM cannot run on Aminer and Web even on a machine with 256GB. VGAE is not a topic model and cannot evaluate topic coherence and perplexity.

Category	Model	Topic Coherence NPMI					Perplexity				
		DS	ML	PL	Aminer	Web	DS	ML	PL	Aminer	Web
Topic models w/o network structure	ProdLDA	10.5±0.3	10.9±0.7	12.1±0.7	8.9±0.0	21.2±0.2	7.97±0.00	7.99±0.00	7.92±0.00	7.60±0.06	8.34±0.00
	DVAE	15.5±0.2	14.7±0.1	15.0±0.1	13.7±0.1	17.6±0.2	14.64±0.15	16.41±0.10	17.52±0.22	20.42±0.25	43.32±0.00
	ETM	7.3±0.2	7.1±0.2	8.7±0.1	5.4±0.3	16.4±0.6	7.92±0.00	7.96±0.00	7.94±0.00	8.31±0.00	8.52±0.00
	WLDA	8.7±0.3	9.8±0.3	11.7±0.4	14.0±1.6	23.9±0.8	17.98±1.88	20.58±0.51	20.60±0.65	14.39±0.25	45.22±0.00
	NSTM	19.0±1.0	17.2±0.7	19.2±0.7	**24.0±0.3**	27.9±0.5	7.80±0.00	7.83±0.00	7.80±0.00	8.26±0.00	8.93±0.00
	GTM	13.0±0.3	18.0±0.5	17.5±0.7	15.2±0.1	21.2±0.9	6.92±0.01	6.97±0.01	6.90±0.00	6.69±0.00	7.84±0.00
Topic models for document networks (LANTM cannot run on large dataset Aminer and Web)	RTM	7.6±0.3	7.1±0.3	9.3±0.2	4.7±0.3	20.9±0.4	7.40±0.03	7.46±0.05	7.52±0.05	7.80±0.00	10.28±0.19
	NRTM	8.2±0.5	9.4±0.1	10.9±0.5	6.1±0.8	26.1±0.3	17.29±0.16	16.94±0.09	16.94±0.07	14.79±0.06	15.09±0.02
	Adjacent-Encoder	12.0±0.2	9.9±0.9	11.3±0.9	3.5±0.5	15.2±0.1	8.06±0.02	7.65±0.05	7.62±0.04	7.17±0.23	8.26±0.01
	LANTM	6.4±0.5	5.4±0.3	7.2±0.8	N.A.	N.A.	8.06±0.02	7.65±0.05	7.62±0.04	N.A.	N.A.
	GTNN	9.9±1.5	7.2±0.6	5.8±0.6	7.6±0.6	7.7±1.7	7.77±0.04	7.75±0.02	7.73±0.01	7.42±0.04	8.13±0.02
	GRTM	13.4±0.7	12.1±0.5	12.6±0.8	14.7±0.2	16.0±1.0	**6.82±0.01**	6.93±0.00	6.88±0.01	6.85±0.00	7.84±0.00
	GNCTM	15.2±0.4	13.2±0.8	13.7±0.5	16.1±0.3	18.8±0.5	7.02±0.16	7.12±0.18	7.11±0.37	7.79±0.48	8.22±0.07
	NetDTM	20.8±0.6	21.1±0.8	19.6±0.3	22.4±0.9	27.4±0.5	7.50±0.03	7.67±0.06	7.65±0.03	6.62±0.01	8.53±0.04
	VGATM	19.9±0.9	17.6±0.4	19.8±1.7	15.9±0.3	22.6±0.5	6.98±0.10	**6.54±0.14**	6.92±0.06	9.05±0.28	8.18±0.21
Our proposed models	DBN	**22.7±0.4**	**21.5±0.6**	**20.9±0.5**	23.6±0.4	28.5±0.2	**6.85±0.00**	6.88±0.00	**6.82±0.00**	**6.49±0.00**	**7.71±0.00**
	D^{2}BN	22.2±0.5	21.2±0.7	20.0±0.3	23.3±0.2	**29.1±0.4**	6.88±0.01	6.91±0.00	6.84±0.00	6.58±0.00	7.89±0.00

TABLE 6: Topic diversity TD (in percentage) at $K = 64$. LANTM cannot run on Aminer and Web even on a machine with 256GB.

Model	Topic Diversity TD				
	DS	ML	PL	Aminer	Web
ProdLDA	81.1±1.1	79.9±1.9	77.9±1.8	86.3±1.2	58.2±0.7
DVAE	47.5±1.0	51.5±0.8	48.1±0.3	74.1±1.5	59.4±0.3
ETM	78.5±1.1	68.4±2.5	75.4±1.0	79.3±0.8	82.7±0.7
WLDA	20.3±0.0	14.8±0.0	17.0±0.0	17.2±0.0	36.6±0.0
NSTM	52.4±1.4	48.6±1.8	50.2±0.8	93.9±0.2	79.0±1.7
GTM	83.6±0.5	83.8±1.0	82.8±0.7	93.2±0.9	82.2±0.5
RTM	78.3±0.7	83.6±1.0	82.9±0.9	93.5±0.5	85.1±0.6
NRTM	15.7±0.3	10.8±0.3	10.9±0.6	20.6±2.5	12.0±1.7
Adjacent-Encoder	73.1±1.3	85.1±1.1	83.9±1.3	91.2±1.0	76.0±0.9
LANTM	73.3±3.0	73.4±1.3	72.2±2.3	N.A.	N.A.
GTNN	51.4±1.6	56.1±1.1	51.8±3.1	68.1±0.7	64.1±1.3
GRTM	81.5±1.1	83.9±1.0	81.8±0.6	92.9±1.0	84.2±2.2
GNCTM	77.8±0.7	83.5±1.1	81.3±1.0	90.5±3.3	81.7±1.3
NetDTM	81.6±1.2	88.1±1.5	83.5±1.5	94.0±0.6	82.8±0.9
VGATM	37.6±2.2	45.0±3.9	42.7±2.3	45.7±0.9	46.2±0.4
DBN	**85.6±0.7**	**89.8±0.4**	**85.8±1.0**	**95.2±0.3**	**85.8±0.2**
D^{2}BN	81.8±0.8	83.3±0.6	77.7±1.4	94.1±1.1	81.3±1.1

topic distributions to predict if there is a link between them. We split the dataset the same as classification. We observe only training documents and links within them for training. After convergence, we infer topics of test documents, and predict links that connect two test documents. As in [29], the probability of a link is $p(e_{ij} = 1|i, j) \propto \exp(-||\mathbf{z}_i - \mathbf{z}_j||^2)$. As in [31], we use AUC as evaluation metric. Table 4(right) presents the results at 64 topics. LANTM cannot run on large datasets Aminer and Web, thus is excluded. Compared to document network models, we highlight that modeling barycenter with auxiliary knowledge can encode similar documents closely. Compared to models without networks, we verify that network indeed brings useful information.

5.2 Topic Analysis

5.2.1 Topic Coherence

An important property of topic model is semantic interpretability, i.e., each document is represented by a topic distribution, and each topic is interpreted by a group of key words. For evaluation, we use topic coherence to test if the key words of a topic coherently reflect the same semantic meaning. Our decoding parameter $2 - \mathbf{C} \in \mathbb{R}^{K\times|\mathcal{V}|}$ is topic-word distribution mentioned in Def. 3.1. Each row is the distribution of a topic over the words, the key words of that topic are the highest values on that row. As in ProdLDA [19], we use NPMI to evaluate the coherence of top-10 key words. We use *Google Web 1T 5-gram Version 1* [52] for evaluation. VGAE is not a topic model and cannot evaluate coherence.

Table 5(left) shows that benefiting from OT, NSTM produces the most coherent topics among baselines. Compared to it, we design OT barycenters to model network structure, thus significantly improve NPMI by 2.1 on average. This is because network helps to capture the situation where two linked documents present different content but consistent semantics. DBN generally performs better than D^{2}BN, since D^{2}BN pays some optimization effort to the reconstruction of adjacency vector, which reduces the precision of content generation. But D^{2}BN is still better than most baselines.

(a) DVAE (b) Adjacent-Encoder (c) DBN (d) D²BN

Fig. 3: T-SNE topic visualization on ML dataset.

TABLE 7: Top-10 key words of 5 randomly selected topics on ML dataset.

Topic	Top-10 key words of DBN
1	optimization, algorithm, trade-off, constraint, scalability, optimal, tradeoff, optimisation, iterative, minimization
2	methodology, theoretical, mathematical, empirical, theory, computational, analysis, modeling, analytical, conceptual
3	multivariate, stochastic, regression, bayesian, nonlinear, parameter, dynamic, gaussian, generalization, inverse
4	generalization, inference, causality, empirical, probabilistic, completeness, causal, causation, first-order, predicate
5	visual, color, image, pattern, subtle, object, eye, contrast, characteristic, optical
Topic	**Top-10 key words of D^{2}BN**
1	finite, topological, symmetric, algebraic, orthogonal, invariant, generalization, topology, subset, discrete
2	feed-forward, multi-layer, feedforward, connectionism, connectionist, two-layer, self-organizing, kohonen, self-organization, network-based
3	processor, microprocessor, parallelism, simd, interface, functionality, hardware, instruction, server, workstation
4	document, copy, text, detailed, instance, specific, read, publication, book, publish
5	genetic, mutation, organism, phenotype, evolution, gene, molecular, evolutionary, trait, protein

5.2.2 Perplexity

A model should generalize well to unseen documents. After training, as in LDA [2], we evaluate perplexity, $\exp\{-\frac{\log \Pr(D_{test})}{\sum_{d' \in D_{test}} N_{d'}}\}$, of 20% test documents. Since perplexity is exponential and varies a lot w.r.t. its power, we instead report its power, $-\frac{\log \Pr(D_{test})}{\sum_{d' \in D_{test}} N_{d'}}$. Lower is better. Table 5(right) shows that our models consistently provide higher likelihood to test documents than baselines, since network helps aggregate text of different documents to alleviate sparsity. Compared to LANTM, we attribute the improvement to Dirichlet prior, which achieves lower reconstruction error than Gaussian, thus providing a better generation to test documents.

5.2.3 Topic Diversity

It is important to evaluate if the discovered K different topics are diverse and not repetitive. Another commonly used metric is topic diversity [6], [26], i.e., TD $= \frac{N_{\text{unique}}}{N_{\text{total}}}$, the percentage of unique key words in the top-10 key words of K topics. TD close to 1 reveals semantically diverse topics. TD close to 0 indicates a large proportion of redundant topics, and other semantically meaningful topics are unexplored, which results in inferior topic discovery.

Table 6 summarizes the results. RTM and Adjacent-Encoder discover the most diverse topics among baselines. Network structure connects similar documents and separates disconnected ones. Modeling network forces topic distributions to focus on each subgroup of connected documents. Different local topology of the network helps improve topic diversity. Compared to RTM and Adjacent-Encoder, we derive more diverse topics, since we use pre-trained embeddings as external knowledge, which capture more diverse and robust semantic meaning. The outperformance over LANTM verifies the advantage of Dirichlet prior.

5.2.4 Topic Interpretability

To intuitively understand what topics our models capture, we randomly select 5 topics and show top-10 key words on ML dataset at Table 7. Most key words coherently reflect the consistent topic. For example, DBN's topic 1 reveals *Constrained Optimization*, and its topic 3 discusses *Multivariate Regression*. For D^{2}BN, topic 2 reflects *Feed-Forward Neural Network*, topic 3 seems *Parallel Processing*, and topic 5 shows *Computational Biology*.

5.2.5 Topic Visualization

To visually understand how topic distributions are learned, we apply t-SNE [53] to project 64-dimensional topic distributions to 2-dimensional visual space, and color documents with their labels. Fig. 3 shows the results on ML. Compared to other three plots, DVAE does not model network structure, thus mixes more documents from different categories. Benefiting from document adjacency, Adjacent-Encoder and DBN present almost similar separation among categories based on visual observation. D^{2}BN produces clearer boundaries, due to double barycentric modeling.

5.3 Model Analysis

5.3.1 Effect of Network Structure

To further verify network structure indeed helps topic modeling, we conduct two experiments. ***i***) We randomly connect documents but keep the number of neighbors each document has. The new network contains wrongly connected neighbors. Fig. 4(a) shows that a random network brings noise on neighborhood and deteriorates the results. This enhances the advantage of a correctly connected network. ***ii***) We randomly remove a proportion of observed links and do document classification using the remaining links. Fig. 4(b) shows results with different percentage of observed links. Compared to the case with no links, more links are observed, better the performance. Links indeed bring useful information, and modeling them improves the quality of learned topic distributions.

5.3.2 Effect of OT Barycenter

Optimal transport barycenter captures document network and incorporates pre-trained embeddings. To test its effect, we remove it from our models and report topic diversity at Fig. 4(c). Both models significantly drop topic diversity without OT barycenter. One potential reason is that OT is good at measuring semantic distance. Given a topic distribution $\mathbf{z}_i$ and word distribution $\mathbf{d}_i$, OT finds the optimal plan between them with less transport cost.

(a) Effect of correct neighbors (b) Percentage of observed links (c) Effect of OT barycenter (d) Effect of Dirichlet prior

(e) Number of convolutional layers (f) Dimension of word embeddings (g) Effect of attention (h) Effect of pre-trained word embedding

Fig. 4: Model analysis on ML dataset.

The minimization of OT barycenter uses as few topics as possible to achieve the transportation between topic $\mathbf{z}_i$ and word $\mathbf{d}_i$. As a result, OT barycenter pushes each topic to focus on its distinct semantics, so that a few diverse topics can still achieve a low-cost transportation. Different topics discuss distinct semantics. The minimization of OT thus produces diverse topics.

5.3.3 Effect of Dirichlet Prior

We conduct two experiments to evaluate the usefulness of our prior. ***i***) We remove Dirichlet and do not impose any prior for topics. ***ii***) Since another widely used prior is Gaussian, we replace Dirichlet with Gaussian. Fig. 4(d) illustrates that Dirichlet prior is indeed useful to discover more diverse topics than the models without prior or with Gaussian. This is because Dirichlet can derive sparse topic distributions, shown by previous works [20]. The sparsity of topic distributions allows each document to focus on only a subset of distinct topics. These subset of topics gradually become specialized in certain semantics, and different subsets of topics preserve unique features. Such process finally produces diverse topics. From Fig. 4(d), we also conclude that having a prior is better than not having it, since the models with Gaussian prior still improve topic diversity. Gaussian prior has zero mean, which forces topics to zero vectors. In contrast, models with no prior do not have this regularization, leading to overlapping topics.

5.3.4 Different Number of Convolutional Layers

We vary the number of convolutional layers L for our encoder, and summarize classification result at Fig. 4(e). When $L = 1$, our model cannot capture higher-order network connectivity, resulting in a low accuracy. After we gradually increase L to 2 and 3, we observe a significant improvement, since structural information is well encoded. However, an overly high value of L influences the result, since more noisy neighbors are encoded.

5.3.5 Dimension of Word Embeddings

We used 300D word embeddings for experiments. To investigate the effect of different dimensions, we do document classification at Fig. 4(f). When we increase the dimension, both models make improvement, since a higher dimension captures more semantic information, boosting the results. D^2BN performs more stably than DBN, because the generation of adjacency vector is auxiliary information and dilutes the effect of word embedding dimensions.

5.3.6 Effect of Attention

To test the effect of attention, we replace the weights with uniform values. Neighbors are equally important. We show the comparison at Fig. 4(g). Removing weights leads to worse results, which verifies the importance of its design. DBN drops more than D^2BN without attention, since adjacency vector in D^2BN represents additional information to offset the influence of uniform attention.

5.3.7 Effect of Pre-trained Word Embeddings

To test word embeddings, we do two experiments. ***i***) We replace pre-trained word embeddings with randomly initialized ones, and optimize them during training. ***ii***) To test if our models can handle our-of-vocabulary (OOV) words, we randomly remove 20% words. Thus, training documents contain OOV words. For OOV words, we check training documents and obtain their contextual words, whose average is the embeddings of OOV words. Fig. 4(h) shows that the models with randomly initialized word embeddings drop the result, since models do not capture auxiliary knowledge. If the corpus has OOV words, the result improves, since models still use partial auxiliary knowledge. The models with OOV words drop the result compared to the models with non-OOV words, since the inferred OOV word embeddings may not be optimal.

6 Conclusion

We propose GNN topic model for document network with OT barycenter. For DBN, we incorporate network structure implicitly by topic barycenter. D^2BN further explicitly pushes topics to double barycenters of semantic and structure spaces. To impose Dirichlet as an optimal transport prior, we use rejection sampling.

ACKNOWLEDGMENTS

This research/project is supported by the Ministry of Education, Singapore under its Tertiary Education Research Fund (MOE Reference Number: MOE2021-TRF-013). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

APPENDIX

Proof. (***i***) We here prove the first inequality, $-\sum_{j\in\mathcal{N}(i)} a_{ij} l(\mathbf{z}_i, \mathbf{d}_j) \geq \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j)$. Its proof is similar to [6]. As a self-contained paper, we provide the details.

Recall that the content decoder is $\phi(\mathbf{z}_i) = \text{softmax}((2-\mathbf{C})^\top \mathbf{z}_i)$. Its denominator is

$$\hat{\phi} = \sum_{w\in\mathcal{V}} \exp(\sum_{k=1}^{K} z_{ik}(2-c_{kw})) = e^2 \sum_{w\in\mathcal{V}} \exp(-\sum_{k=1}^{k} z_{ik} c_{kw}). \tag{18}$$

The log-likelihood of content generation between document i and its neighbors $j \in \mathcal{N}(i)$ is

$$\begin{aligned}
&\sum_{j\in\mathcal{N}(i)} a_{ij} l(\mathbf{z}_i, \mathbf{d}_j) = \sum_{j\in\mathcal{N}(i)} \frac{a_{ij}}{\sum_{w'\in\mathcal{V}} n_{j,w'}} \sum_{w\in\mathcal{V}} n_{j,w} \log \phi(\mathbf{z}_i)_w \\
&= \sum_{j\in\mathcal{N}(i)} \frac{a_{ij}}{\sum_{w'\in\mathcal{V}} n_{j,w'}} \sum_{w\in\mathcal{V}} n_{j,w} (\sum_{k=1}^{K} z_{ik}(2-c_{kw}) - \log\hat{\phi}) \\
&= \sum_{j\in\mathcal{N}(i)} a_{ij} \left(2 - \log\hat{\phi} - \frac{1}{\sum_{w'\in\mathcal{V}} n_{j,w'}} \sum_{w\in\mathcal{V}} n_{j,w} \sum_{k=1}^{K} z_{ik} c_{kw}\right)
\end{aligned} \tag{19}$$

Since transport plan $\mathbf{T} \in U(\mathbf{z}_i, \mathbf{d}_j) = \{\mathbf{T} \in \mathbb{R}_+^{K\times|\mathcal{V}|} | \mathbf{T}\mathbf{1}_{|\mathcal{V}|} = \mathbf{z}_i, \mathbf{T}^\top \mathbf{1}_K = \mathbf{d}_j\}$ has $\mathbf{z}_i$ and $\mathbf{d}_j$ as maginals, we define $t_{kw} = p_{kw} d_{jw}$. Here we introduce another conditional transport plan $\mathbf{P} \in U'(\mathbf{z}_i, \mathbf{d}_j) = \{\mathbf{P} \in \mathbb{R}_+^{K\times|\mathcal{V}|} | \sum_{w\in\mathcal{V}} p_{kw} d_{jw} = z_{ik}, \sum_{k=1}^{K} p_{kw} = 1\}$. $d_{jw} = \frac{n_{jw}}{\sum_{w'\in\mathcal{V}} n_{jw'}}$ is the normalized word count of word w in document j. With these definitions, we have

$$\begin{aligned}
&\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) = \sum_{j\in\mathcal{N}(i)} a_{ij} \min_{\mathbf{P}\in U'(\mathbf{z}_i,\mathbf{d}_j)} \sum_{w\in\mathcal{V}} \sum_{k=1}^{K} p_{kw} d_{jw} c_{kw} \\
&= \sum_{j\in\mathcal{N}(i)} \frac{a_{ij}}{\sum_{w'\in\mathcal{V}} n_{jw'}} \min_{\mathbf{P}\in U'(\mathbf{z}_i,\mathbf{d}_j)} \sum_{w\in\mathcal{V}} n_{jw} \sum_{k=1}^{K} p_{kw} c_{kw}.
\end{aligned} \tag{20}$$

If we set $p_{kw} = z_{ik}$, we discover that $\mathbf{P}$ satisfies the requirement of $U'(\mathbf{z}_i, \mathbf{d}_j)$. Since $p_{kw} = z_{ik}$ may not be the optimal solution, we have

$$\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \leq \sum_{j\in\mathcal{N}(i)} \frac{a_{ij}}{\sum_{w'\in\mathcal{V}} n_{jw'}} \sum_{w\in\mathcal{V}} n_{jw} \sum_{k=1}^{K} z_{ik} c_{kw}. \tag{21}$$

Taking Eq. 19 and 21 together, given $|\mathcal{V}| \geq 8$, we have

$$\begin{aligned}
&-\sum_{j\in\mathcal{N}(i)} a_{ij} l(\mathbf{z}_i, \mathbf{d}_j) = -\sum_{j\in\mathcal{N}(i)} a_{ij} \Big(2 - \log\hat{\phi} \\
&\qquad - \frac{1}{\sum_{w'\in\mathcal{V}} n_{j,w'}} \sum_{w\in\mathcal{V}} n_{j,w} \sum_{k=1}^{K} z_{ik} c_{kw}\Big) \\
&\geq -2 + \log\hat{\phi} + \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \\
&= \log \sum_{w\in\mathcal{V}} \exp(-\sum_{k=1}^{k} z_{ik} c_{kw}) + \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \\
&\geq \log(|\mathcal{V}|) - 2 + \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \\
&\geq \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j).
\end{aligned} \tag{22}$$

(***ii***) We here prove the second inequality, $\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \geq \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_j, \mathbf{d}_j)$.

We apply the proof by contradiction. Given $\mathbf{C}$ and a_{ij}, we assume $\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) < \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_j, \mathbf{d}_j)$ is correct, which tells us that the solution Eq. 23 is better than Eq. 24. Here we use $\mathbf{z}_j^*$ to represent the best solution of OT $d_{\mathbf{C}}(\cdot, \mathbf{d}_j)$.

$$\mathbf{z}_1^* = \mathbf{z}_2^* = ... = \mathbf{z}_{|\mathcal{N}(i)|}^* = \mathbf{z}_i \tag{23}$$

$$\mathbf{z}_1^* = \mathbf{z}_1, \quad \mathbf{z}_2^* = \mathbf{z}_2, \quad ..., \quad \mathbf{z}_{|\mathcal{N}(i)|}^* = \mathbf{z}_{|\mathcal{N}(i)|} \tag{24}$$

Eq. 23 shows that all the best solutions are the same, while Eq. 24 shows that each individual OT has its own solution and different OTs may produce different solutions. However, given that $\mathbf{z}_j = \arg\min_{\mathbf{z}_j} d_{\mathbf{C}}(\mathbf{z}_j, \mathbf{d}_j)$ is already the best solution of its individual OT, other solutions, such as $\mathbf{z}_j^* = \mathbf{z}_i$, may not produce the minimum OT distance. Equivalently, the assumption of $\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) < \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_j, \mathbf{d}_j)$ contradicts the given information.

Based on the analysis above, we have

$$d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \geq d_{\mathbf{C}}(\mathbf{z}_j, \mathbf{d}_j). \tag{25}$$

Since cost matrix $\mathbf{C}$ and weights a_{ij} are given, taking all the neighbors $j \in \mathcal{N}(i)$ together, we have

$$\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \geq \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_j, \mathbf{d}_j). \tag{26}$$

□

REFERENCES

[1] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[3] J. Chang and D. Blei, "Relational topic models for document networks," in *Artificial intelligence and statistics*. PMLR, 2009, pp. 81–88.

[4] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in neural information processing systems*, vol. 26, pp. 2292–2300, 2013.

[5] V. Huynh, H. Zhao, and D. Phung, "Otlda: A geometry-aware optimal transport approach for topic modeling," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[6] H. Zhao, D. Phung, V. Huynh, T. Le, and W. Buntine, "Neural topic model via optimal transport," in *International Conference on Learning Representations*, 2020.

[7] M. Cuturi and A. Doucet, "Fast computation of wasserstein barycenters," in *International conference on machine learning*. PMLR, 2014, pp. 685–693.

[8] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ICLR*, 2016.

[9] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[10] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 2609–2615.

[11] J. Li, J. Yu, J. Li, H. Zhang, K. Zhao, Y. Rong, H. Cheng, and J. Huang, "Dirichlet graph variational autoencoder," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5274–5283, 2020.

[12] L. Yang, J. Gu, C. Wang, X. Cao, L. Zhai, D. Jin, and Y. Guo, "Toward unsupervised graph neural network: Interactive clustering and embedding via optimal transport," in *2020 IEEE international conference on data mining (ICDM)*. IEEE, 2020, pp. 1358–1363.

[13] J. Zhang, X. Xiao, L.-K. Huang, Y. Rong, and Y. Bian, "Fine-tuning graph neural networks via graph topology induced optimal transport," 2022.

[14] L. Chen, G. Wang, C. Tao, D. Shen, P. Cheng, X. Zhang, W. Wang, Y. Zhang, and L. Carin, "Improving textual network embedding with global attention via optimal transport," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5193–5202.

[15] X. Luo, Y. Zhao, Y. Qin, W. Ju, and M. Zhang, "Towards semi-supervised universal graph classification," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[16] I. Vayansky and S. A. Kumar, "A review of topic modeling methods," *Information Systems*, vol. 94, p. 101582, 2020.

[17] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[18] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *International conference on machine learning*. PMLR, 2016, pp. 1727–1736.

[19] A. Srivastava and C. Sutton, "Autoencoding variational inference for topic models," *ICLR*, 2017.

[20] S. Burkhardt and S. Kramer, "Decoupling sparsity and smoothness in the dirichlet variational autoencoder topic model." *Journal of Machine Learning Research*, vol. 20, no. 131, pp. 1–27, 2019.

[21] L. Yang, F. Wu, J. Gu, C. Wang, X. Cao, D. Jin, and Y. Guo, "Graph attention topic modeling network," in *Proceedings of The Web Conference 2020*, 2020, pp. 144–154.

[22] Q. Zhu, Z. Feng, and X. Li, "Graphbtm: Graph enhanced autoencoded variational inference for biterm topic model," in *Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, 2018.

[23] R. Das, M. Zaheer, and C. Dyer, "Gaussian lda for topic models with word embeddings," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 795–804.

[24] A. B. Dieng, F. J. Ruiz, and D. M. Blei, "Topic modeling in embedding spaces," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 439–453, 2020.

[25] H. Xu, W. Wang, W. Liu, and L. Carin, "Distilled wasserstein learning for word embedding and topic modeling," *Advances in Neural Information Processing Systems*, 2018.

[26] F. Nan, R. Ding, R. Nallapati, and B. Xiang, "Topic modeling with wasserstein autoencoders," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 6345–6381.

[27] H. Bai, Z. Chen, M. R. Lyu, I. King, and Z. Xu, "Neural relational topic models for scientific article analysis," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 27–36.

[28] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[29] C. Zhang and H. W. Lauw, "Topic modeling on document networks with adjacent-encoder," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6737–6745.

[30] Q. Xie, J. Huang, P. Du, M. Peng, and J.-Y. Nie, "Graph topic neural network for document representation," in *Proceedings of the Web Conference 2021*, 2021, pp. 3055–3065.

[31] Y. Wang, X. Li, and J. Ouyang, "Layer-assisted neural topic modeling over document networks." in *IJCAI*, 2021, pp. 3148–3154.

[32] W. Wang, Z. Gong, J. Ren, F. Xia, Z. Lv, and W. Wei, "Venue topic model–enhanced joint graph modelling for citation recommendation in scholarly big data," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 20, no. 1, pp. 1–15, 2020.

[33] X. Huang and X. Liu, "Incorporating a topic model into a hypergraph neural network for searching-scenario oriented recommendations," *Applied Sciences*, vol. 12, no. 15, p. 7387, 2022.

[34] D. C. Zhang and H. W. Lauw, "Topic modeling for multi-aspect listwise comparisons," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 2507–2516.

[35] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in neural information processing systems*, vol. 33, pp. 9912–9924, 2020.

[36] W. Chang, Y. Shi, H. Tuan, and J. Wang, "Unified optimal transport framework for universal domain adaptation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 29 512–29 524, 2022.

[37] O. K. Shirekar, A. Singh, and H. Jamali-Rad, "Self-attention message passing for contrastive few-shot learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 5426–5436.

[38] C. Naesseth, F. Ruiz, S. Linderman, and D. Blei, "Reparameterization gradients through acceptance-rejection sampling algorithms," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 489–498.

[39] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, 2013.

[40] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[41] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[42] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Information Retrieval*, vol. 3, no. 2, pp. 127–163, 2000.

[43] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 990–998.

[44] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 497–506.

[45] D. Zhou, X. Hu, and R. Wang, "Neural topic modeling by incorporating document relationship graph," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 3790–3796.

[46] Q. Xie, J. Huang, P. Du, and M. Peng, "Graph relational topic model with higher-order graph attention auto-encoders," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 2604–2613.

[47] Q. Xie, Y. Zhu, J. Huang, P. Du, and J.-Y. Nie, "Graph neural collaborative topic model for citation recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 40, no. 3, pp. 1–30, 2021.

[48] D. C. Zhang and H. Lauw, "Dynamic topic models for temporal document networks," in *International Conference on Machine Learning*. PMLR, 2022, pp. 26 281–26 292.

[49] D. C. Zhang and H. W. Lauw, "Variational graph author topic modeling," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2429–2438.

[50] S. Lin, C. Liu, P. Zhou, Z.-Y. Hu, S. Wang, R. Zhao, Y. Zheng, L. Lin, E. Xing, and X. Liang, "Prototypical graph contrastive learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[51] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *The Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.

[52] S. Evert, "Google web 1t 5-grams made easy (but not for the computer)," in *Proceedings of the NAACL HLT 2010 Sixth Web as Corpus Workshop*, 2010, pp. 32–40.

[53] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

Delvin Ce Zhang is a PhD Candidate in Computer Science at Singapore Management University, Singapore. He received Bachelor's Degree in Computer Science and Technology at Sichuan University, P. R. China, in 2018. His research interests are in graph neural networks, graph representation learning, text mining, etc. His first-authored papers are published on NeurIPS, ICML, KDD, AAAI, CIKM, ECMLP-KDD, etc.

Hady W. Lauw is an associate professor at SMU, where he directs the BSc (Computer Science) programme and leads the Preferred.AI research group working on modeling preferences and recommender systems. Formerly, he served as postdoctoral researcher at Microsoft Research in Silicon Valley, as well as scientist at A*STAR's Institute for Infocomm Research. Earlier, he received his PhD from Nanyang Technological University.