Variational Graph Author Topic Modeling

Delvin Ce Zhang
Singapore Management University
Singapore
cezhang.2018@smu.edu.sg

Hady W. Lauw
Singapore Management University
Singapore
hadywlauw@smu.edu.sg

ABSTRACT

While Variational Graph Auto-Encoder (VGAE) has presented promising ability to learn representations for documents, most existing VGAE methods do not model a latent topic structure and therefore lack *semantic interpretability*. Exploring hidden topics within documents and discovering key words associated with each topic allow us to develop a semantic interpretation of the corpus. Moreover, documents are usually associated with authors. For example, news reports have journalists specializing in writing certain type of events, academic papers have authors with expertise in certain research topics, etc. Modeling *authorship information* could benefit topic modeling, since documents by the same authors tend to reveal similar semantics. This observation also holds for documents published on the same *venues*. However, most topic models ignore the auxiliary authorship and publication venues. Given above two challenges, we propose a Variational Graph Author Topic Model for documents to integrate both semantic interpretability and authorship and venue modeling into a unified VGAE framework. For authorship and venue modeling, we construct a hierarchical multi-layered document graph with both intra- and cross-layer topic propagation. For semantic interpretability, three word relations (contextual, syntactic, semantic) are modeled and constitute three word sub-layers in the document graph. We further propose three alternatives for variational divergence. Experiments verify the effectiveness of our model on supervised and unsupervised tasks.

CCS CONCEPTS

• **Information systems → Data mining**; **Document topic models**; • **Computing methodologies → Topic modeling**.

KEYWORDS

Graph Neural Networks; Variational Graph Auto-Encoder; Author Topic Modeling; Text Mining

ACM Reference Format:
Delvin Ce Zhang and Hady W. Lauw. 2022. Variational Graph Author Topic Modeling. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3534678.3539310

1 INTRODUCTION

Due to the explosion of documents, there is a need to automatically organize overwhelmed corpus. One effective method is to infer low-dimensional document representations, which could fulfill real-world tasks, e.g., document classification [39]. Recently, Variational Graph Auto-Encoder (VGAE) [12] has presented promising ability to learn effective document representations. However, when modeling documents, we usually assume a latent topic structure [2]. Each document is represented by a topic distribution, each topic is interpreted by its key words. Such topic structure offers *semantic interpretability* and allows us to better understand the main theme of the corpus. However, most VGAE methods do not model the notion of topics, leading to uninterpretable representations.

As an important statistical tool for exploratory analysis of text corpora, topic model allows us to explore latent topics within documents. Moreover, a document is usually associated with authors. For example, news reports have journalists specializing in writing a certain category of events; scientific papers have authors with expertise in certain research topics. Modeling authors could benefit topic model, since documents by the same authors reveal similar semantics, and authorship could connect these documents and jointly infer their topics. This observation also holds for venues, e.g., papers from the same journal exhibit similar research areas. However, traditional topic models, e.g., LDA [2], infer topics based on plain text only, without auxiliary *authorship* or *venues*. Recently, Author Topic Models [28] are proposed for authorship and venue modeling.

Challenges. Most existing graph neural networks for text embedding, e.g., TextGCN [39], lack topic modeling, leading to uninterpretable representations. Although there exist a few studies [6, 34] modeling the concept of topics, topics are learned in advance by existing models to construct the graph, independently from graph convolution. In contrast, our proposed model integrates both VGAE and topic modeling into a unified architecture where the learned topic proportions of documents enjoy semantic interpretability.

Some works recognize the value of topic modeling. However, models, e.g., LDA [2] and the recent GATON [38], ignore authorship and venues of documents. Authorship and venues indicate semantic similarities, and modeling them could uncover meaningful topics.

Author topic models, e.g., ATM [28] and ACT [30], consider authorship and venues. However, they mainly infer topics for authors and fail to also learn topics for documents. As a result, automatically organizing documents, e.g., classification, remains unsolved.

Approach. Motivated by above challenges, we design **V**ariational **G**raph **A**uthor **T**opic **M**odel (VGATM) to achieve both semantic interpretability and authorship (venue) modeling. Specifically, we extend VGAE and unify it with topic modeling. For authorship and venue modeling, we design a document layer, an author layer, and a venue layer, and construct a hierarchical multi-layered document graph as the corpus. For semantic interpretability, we model three

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539310

word relations (contextual, syntactic, and semantic) as three word sub-layers. Topics are propagated both within each layer to capture graph structure and across different layers for semantic learning.

In addition, we also investigate the variational divergence term in our model, which acts as the prior. We propose three alternatives: *i*) Gaussian prior with KL divergence; *ii*) Dirichlet prior with KL divergence; and *iii*) Gaussian prior with Wasserstein distance.

Contributions. First, we propose VGATM unifying VGAE and topic model to jointly achieve semantic interpretability and authorship modeling. Our model also accommodates publication venues of documents. For semantic interpretability, we construct a three word sub-layers to describe contextual, syntactic, and semantic word relations. Second, to model authorship and venues, we design a hierarchical multi-layered document graph, and simulate intra- and cross-layer topic propagation to integrate auxiliary data into documents' topic proportions. Third, we propose three design alternatives for variation divergence to improve topic modeling.

2 RELATED WORK

Graph neural networks are designed to learn vertex representations. Variational Graph Auto-Encoder (VGAE) [12] extends VAE [10] where Graph Convolutional Network (GCN) [11] is the vertex encoder. ARVGA [26] improves VGAE by adversarial training. DGVAE [15] replaces Gaussian prior with Dirichlet. Graphite [8] extends the decoder of VGAE by an iterative graph refinement strategy. CGVAE [19] investigates the application in chemistry. MLHNE [41] constructs a multi-layered graph. All these models are not topic models and lack semantic interpretability.

To explore the latent topics within documents, topic models are proposed. LDA [2] is a graphical model. Recently, neural topic models attract more attention. NVDM [23] extends VAE for topic modeling. ProdLDA [29] and DVAE [3] design Dirichlet prior, WHAI [43] uses Gamma prior. WLDA [25] applies Wasserstein distance in the word space. More recently, topic models are based on graph structure. GATON [38] designs a bipartite graph for topic propagation. GraphBTM [45] improves biterm topic model using graphs. DHTG [34] and HyperGAT [6] use existing topic models to construct graphs. They fail to consider auxiliary authors and venues.

Author Topic Model (ATM) [28] derives topics for authors. ACT [30] improves ATM by modeling venues. CAT [31] further models paper citations. They do not infer topics for documents. CNTM [16] infers topics for both documents and authors, but fails to consider venues. There are topic models for document graphs [1, 4, 40, 42]. They consider first-order neighbors only. Recently, higher-order adjacency is considered [33, 35]. They are proposed for the direct connection (citation), and ignore authorship and venues.

Previously, text classification models are based on CNN [9] and RNN [18]. Recently, graph models present promising ability, e.g., TextGCN [39], TensorGCN [20], HGAT [17], etc. They are not topic models and fail to consider authorship or venues. The recent TVGAE [36] models topics, but still ignores authorship and venues.

3 PRELIMINARIES

Here, we introduce preliminaries. Table 1 summarizes notations.

We are given a corpus of documents $C = \{\mathcal{D}, \mathcal{A}, \mathcal{V}, \mathcal{X}\}$ with authors and venues. $\mathcal{D} = \{\mathbf{d}_i\}$ is a set of documents. Each document d contains N_d words in the vocabulary $\mathcal{W}$, i.e., $\mathbf{d} = \{w_{d,n}\}_{n=1}^{N_d} \subseteq \mathcal{W}$. Document d has a sequence of A_d authors $\mathbf{a}_d = \{a_{d,n}\}_{n=1}^{A_d} \subseteq \mathcal{A}$ and a venue $v_d \in \mathcal{V}$. Besides, we also observe edges $\mathcal{X}$ connecting documents, such as citations between papers. $x_{d_i,d_j} = 1$ if there is an edge between d_i and d_j, otherwise $x_{d_i,d_j} = 0$. We model undirected edges, $x_{d_i,d_j} = x_{d_j,d_i}$. We will use *edge* and *link* interchangeably. As in [33], when no edges $\mathcal{X}$ are observed, we induce κNN edges using documents' content similarity. We include $\mathcal{X}$ because we will use it to construct a document graph for author topic modeling.

Table 1: Summary of math notations.

Notation	Description
C	a corpus
$\mathcal{D}$	a set of documents
$\mathcal{W}$	vocabulary
$\mathcal{A}$	a set of authors
$\mathcal{V}$	a set of publication venues
$\mathcal{X}$	edge connections among documents
$\mathcal{G}$	a hierarchical multi-layered document graph based on corpus C
$\mathcal{U}$	a set of vertices of $\mathcal{G}$, we have $\mathcal{U} = \mathcal{D} \cup \mathcal{W} \cup \mathcal{A} \cup \mathcal{V}$
$\mathcal{E}$	a set of edges of $\mathcal{G}$, we have $\mathcal{X} \subseteq \mathcal{E}$
$\mathcal{O}$	a set of vertex types
$\mathcal{T}$	a set of edge types
K	number of topics
$q(\mathbf{z}_i)$	variational posterior distribution of vertex i, parameterized by our encoder
$\log p(\cdot\|\cdot)$	log-likelihood of generation, or reconstruction term
$p(\mathbf{z})$	predefined prior distribution
$\mathcal{R}$	divergence metric
$\tilde{\mathbf{z}}_i^{(l)}$	the representation after linear projection at the l-th convolutional step
$\mathbf{z}_i^{(l)}$	topic proportion of vertex i output by the l-th convolutional step at Eq. 9
$\tilde{\mathbf{h}}_w^{(L)}$	topic proportion representing document d's whole content at Eq. 13
M	number of negative samples at Eq. 25

Given C, a corpus of documents with auxiliary authors and venues, as input, our goal is to output topic proportions for $|\mathcal{D}|$ documents to preserve textual content $\mathcal{D}$, authorship $\mathcal{A}$, and venues $\mathcal{V}$ where we use edge connections $\mathcal{X}$ as assisted graph structure.

Definition 3.1 (Variational Graph Auto-Encoder (VGAE)). *Given documents $\mathcal{D}$ and a graph structure $\mathcal{X}$ as inputs, VGAE learns a mapping function q to project documents to K-dimensional embedding space by $q(\mathbf{Z}|\mathcal{D}, \mathcal{X}) \in \mathbb{R}^{|\mathcal{D}| \times K}$, preserving content $\mathcal{D}$ and graph structure $\mathcal{X}$. VGAE aims to maximize the following objective.*

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{Z}|\mathcal{D},\mathcal{X})} \log p(\mathcal{X}|\mathbf{Z}) - \mathcal{R}[q(\mathbf{Z}|\mathcal{D}, \mathcal{X}) || p(\mathbf{Z})]. \tag{1}$$

Encoder $q(\mathbf{Z}|\mathcal{D}, \mathcal{X})$ is variational posterior parameterized by a Graph Convolutional Network [11]. Decoder is log-likelihood $\log p(\mathcal{X}|\mathbf{Z})$ reconstructing the graph structure. Divergence $\mathcal{R}$ pushes variational posterior to a predefined prior $p(\mathbf{Z})$. VGAE uses KL divergence as $\mathcal{R}$.

In this paper, we will extend VGAE as a topic model and incorporate auxiliary authorship $\mathcal{A}$ and publication venues $\mathcal{V}$.

Definition 3.2 (Wasserstein Distance). *Wasserstein distance is a metric to measure the distance between two probability distributions. Let $\mathcal{P}(\mathbb{R}^K)$ be the set of Borel probability measures on K-dimensional space $\mathbb{R}^K$. For $\rho \geq 1$, and two K-dimensional probability measures $\boldsymbol{u}$ and $\boldsymbol{v}$ in $\mathcal{P}(\mathbb{R}^K)$, their ρ-Wasserstein distance is*

$$W_\rho(\boldsymbol{u}, \boldsymbol{v}) = \left(\inf_{\pi \in \Pi(\boldsymbol{u}, \boldsymbol{v})} \int_{\mathbb{R}^K \times \mathbb{R}^K} ||x - y||^\rho d\pi(x, y) \right)^{1/\rho}. \tag{2}$$

Here, $\Pi(\boldsymbol{u}, \boldsymbol{v})$ is the set of all probability measures on $\mathbb{R}^K \times \mathbb{R}^K$ with $\boldsymbol{u}$ and $\boldsymbol{v}$ as marginal distributions.

We will investigate the effect of Wasserstein distance as the alternative of KL divergence for prior regularization in our model.

4 HIERARCHICAL MULTI-LAYERED GRAPH

Given C, to design graph convolution to obtain topic proportions of documents, we need to construct a document graph using C. Below we first define a *multi-layered* graph. Then we extend it to a *hierarchical multi-layered* structure. See Fig. 1(a) for an overview.

4.1 Multi-Layered Document Graph

Considering author and document as vertices, we connect authors and documents with authorship edges. Similarly, for documents' words and venues, edges are contents and publications, respectively. Formally, a multi-layered document graph $\mathcal{G} = \{\mathcal{U}, \mathcal{E}, \mathcal{O}, \mathcal{T}\}$ consists of a vertex set $\mathcal{U}$ and an edge set $\mathcal{E}$, and is associated with two mapping functions θ and ϑ. The vertex mapping function $\theta : \mathcal{U} \to \mathcal{O}$ projects each vertex $i \in \mathcal{U}$ to a specific type $o \in \mathcal{O} = \{\text{document, word, author, venue}\}$. Each type o corresponds to a *graph layer* containing vertices of the same type. The edge mapping function $\vartheta : \mathcal{E} \to \mathcal{T}$ projects edge e_{ij} between vertices i and j to an edge type $t \in \mathcal{T}$={document-word, document-author, document-venue}. These three types are cross-layer edges.

We further construct four types of intra-layer edges: document-document, author-author, venue-venue, and word-word. Edges between documents $\mathcal{X}$ defined above can be citations between academic papers, hyperlinks between Web pages, or κNN edges based on documents' content similarity. Author-author edges are collaboration co-authorship. We do not discover appropriate methods for venue-venue edges, we simply add self-loop edges for venues. We will define word-word edges shortly. Thus, there are $|\mathcal{O}| = 4$ graph layers. $\mathcal{U} = \mathcal{D} \cup \mathcal{W} \cup \mathcal{A} \cup \mathcal{V}$ and $\mathcal{X} \subseteq \mathcal{E}$. Fig. 1(a) contains 4 graph layers, black and green edges are intra- and cross-layer edges.

4.2 Three Word Sub-Layers

We now define word-word edges. As shown by topic model literature [5], word co-occurrence has a significant impact on topic interpretability. In our model, word-word edges depict the co-occurred connections. Thus, to improve topic quality, we build word-word edges using three word relations, i.e., contextual, syntactic, and semantic, which extend the word layer above to be three sub-layers.

Contextual word sub-layer describes the local co-occurrence of words within the corpus. Following [39], we use point-wise mutual information (PMI) to capture contextual relation with a fixed-size sliding window strategy. We slide the window on a sequence of words within the corpus to obtain *contextual co-occurrence relation*, after which, for each pair of words (w_i, w_j), we calculate PMI score.

$$S_{ctx}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}. \tag{3}$$

$p(w_i, w_j)$ is the probability of word pair (w_i, w_j) co-occurring in the same sliding window, and $p(w_i)$ and $p(w_j)$ represent the probability of respective word occurring in a sliding window. We estimate $p(w_i, w_j) = \frac{N_{ctx}(w_i,w_j)}{N_{ctx}}$ and $p(w_i) = \frac{N_{ctx}(w_i)}{N_{ctx}}$. $N_{ctx}(w_i, w_j)$ is the number of co-occurrences of word pair (w_i, w_j) across all sliding windows, and $N_{ctx}(w_i)$ and $N_{ctx}(w_j)$ are similarly defined for a single word w_i and w_j, respectively. N_{ctx} is the total number of sliding windows. After calculating PMI scores for all pairs of words, for each word, we select its top-5 PMI scores as its neighboring words and construct edges as contextual co-occurrence relation.

Syntactic word sub-layer represents the syntactic dependency relation between words. Following [20], we use Stanford CoreNLP parser [21] to extract dependency between words. For each pair of words (w_i, w_j), we calculate *syntactic co-occurrence* score by

$$S_{syn} = \frac{N_{syn}(w_i, w_j)}{N_{\textit{co-occur}}(w_i, w_j)}. \tag{4}$$

$N_{syn}(w_i, w_j)$ is the number of times that word pair (w_i, w_j) presents syntactic dependency, which is normalized by $N_{\textit{co-occur}}(w_i, w_j)$, the number of total co-occurrences of (w_i, w_j). For each word, its top-5 syntactic scores denote its syntactic co-occurrence neighbors.

Semantic word sub-layer connects words with similar semantic meaning, captured by pretrained word embeddings [27]. For each pair (w_i, w_j), we calculate *semantic co-occurrence* score.

$$S_{sem} = \cos(g(w_i), g(w_j)). \tag{5}$$

$g(w_i)$ and $g(w_j)$ respectively denotes the word embedding of w_i and w_j. $\cos(\cdot, \cdot)$ is cosine similarity. Again, for each word, the top-5 semantically related words are its neighbors as semantic relation.

In Fig. 1(a), three sub-layers of words share the same set of vertices, i.e., words, but the edge connections are different, since different co-occurrence relations link different words as neighbors. Encapsulating three sub-layers of words into above multi-layered document graph, we obtain a *hierarchical* multi-layered structure.

5 METHODOLOGY

We introduce **V**ariational **G**raph **A**uthor **T**opic **M**odel (VGATM), extending VGAE as a topic model with auxiliary authors and venues.

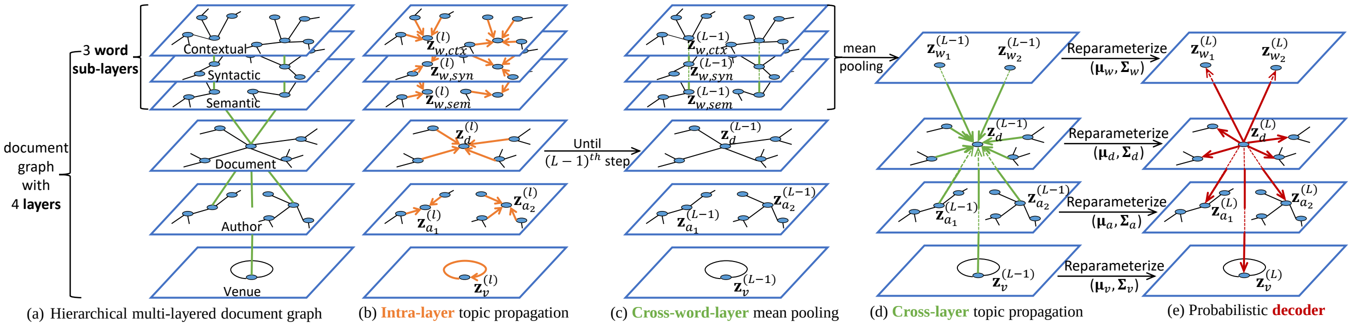
As an overview, we describe the generative process of VGATM. Following LDA, given a corpus C, we generate observations: content $\mathcal{D}$, authors $\mathcal{A}$, venues $\mathcal{V}$, and edges between documents $\mathcal{X}$.

(1) For each word $w \in \mathcal{W}$, author $a \in \mathcal{A}$, and venue $v \in \mathcal{V}$:
   (a) Draw K-dimensional topic proportion $\mathbf{z}_w \sim p(\mathbf{z}_w)$, $\mathbf{z}_a \sim p(\mathbf{z}_a)$, and $\mathbf{z}_v \sim p(\mathbf{z}_v)$.
(2) For each document $d \in \mathcal{D}$:
   (a) Draw K-dimensional topic proportion $\mathbf{z}_d \sim p(\mathbf{z}_d)$.
   (b) Draw each word $w_{d,n} \sim p(w_{d,n}|\mathbf{z}_d, \mathbf{z}_{w_{d,n}})$, $n = 1, 2, ..., N_d$.
   (c) Draw each author $a_{d,n} \sim p(a_{d,n}|\mathbf{z}_d, \mathbf{z}_{a_{d,n}})$, $n = 1, 2, ..., A_d$.
   (d) Draw a venue $v_d \sim p(v_d|\mathbf{z}_d, \mathbf{z}_{v_d})$.
   (e) If d's label y_d exists, draw a label $y_d \sim p(y_d|\mathbf{z}_d)$.
(3) For each pair of documents d_i and d_j where $d_i, d_j \in \mathcal{D}$:
   (a) Draw an edge indicator $x_{d_i,d_j} \sim p(x_{d_i,d_j}|\mathbf{z}_{d_i}, \mathbf{z}_{d_j})$.

Maximizing log-likelihood $\mathcal{L}(C)$ is intractable, as in VGAE [12], we instead maximize its evidence lower bound below.

$$\begin{aligned}\mathcal{L} = \mathbb{E}_{q(\mathbf{Z}_\mathcal{D},\mathbf{Z}_\mathcal{W},\mathbf{Z}_\mathcal{A},\mathbf{Z}_\mathcal{V})}\Big(\sum_{d\in\mathcal{D}}[\log p(\mathbf{d}|\mathbf{z}_d, \mathbf{Z}_\mathcal{W}) + \log p(\mathbf{a}_d|\mathbf{z}_d, \mathbf{Z}_\mathcal{A})\\ + \log p(\mathbf{v}_d|\mathbf{z}_d, \mathbf{Z}_\mathcal{V}) + \lambda_{label} \log p(\mathbf{y}_d|\mathbf{z}_d)] + \sum_{d_i,d_j\in\mathcal{D}} \log p(x_{d_i,d_j}|\mathbf{z}_{d_i}, \mathbf{z}_{d_j})\Big)\\ - \lambda_{prior}(\mathcal{R}[q(\mathbf{Z}_\mathcal{D})||p(\mathbf{Z})] + \mathcal{R}[q(\mathbf{Z}_\mathcal{W})||p(\mathbf{Z})] + \mathcal{R}[q(\mathbf{Z}_\mathcal{A})||p(\mathbf{Z})]\\ + \mathcal{R}[q(\mathbf{Z}_\mathcal{V})||p(\mathbf{Z})]).\end{aligned} \tag{6}$$

We use upper letter $\mathbf{Z}_\mathcal{D} \in \mathbb{R}^{|\mathcal{D}|\times K}$ as a collection of latent topics of all the documents, and ditto for $\mathbf{Z}_\mathcal{W}, \mathbf{Z}_\mathcal{A}, \mathbf{Z}_\mathcal{V}$. K is the number of


3 word sub-layers
Contextual
Syntactic
Semantic
document graph with 4 layers
Document
Author
Venue
$\mathbf{z}_{w,ctx}^{(l)}$
$\mathbf{z}_{w,syn}^{(l)}$
$\mathbf{z}_{w,sem}^{(l)}$
$\mathbf{z}_{d}^{(l)}$
$\mathbf{z}_{a_1}^{(l)}$
$\mathbf{z}_{a_2}^{(l)}$
$\mathbf{z}_{v}^{(l)}$
Until $(L-1)^{th}$ step
$\mathbf{z}_{w,ctx}^{(L-1)}$
$\mathbf{z}_{w,syn}^{(L-1)}$
$\mathbf{z}_{w,sem}^{(L-1)}$
$\mathbf{z}_{d}^{(L-1)}$
$\mathbf{z}_{a_1}^{(L-1)}$
$\mathbf{z}_{a_2}^{(L-1)}$
$\mathbf{z}_{v}^{(L-1)}$
mean pooling
$\mathbf{z}_{w_1}^{(L-1)}$
$\mathbf{z}_{w_2}^{(L-1)}$
Reparameterize $(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$
Reparameterize $(\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d)$
Reparameterize $(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a)$
Reparameterize $(\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)$
$\mathbf{z}_{w_1}^{(L)}$
$\mathbf{z}_{w_2}^{(L)}$
$\mathbf{z}_{d}^{(L)}$
$\mathbf{z}_{a_1}^{(L)}$
$\mathbf{z}_{a_2}^{(L)}$
$\mathbf{z}_{v}^{(L)}$

(a) Hierarchical multi-layered document graph (b) **Intra-layer** topic propagation (c) **Cross-word-layer** mean pooling (d) **Cross-layer** topic propagation (e) Probabilistic **decoder**

Figure 1: Model architecture. (a) Given a corpus with auxiliary authors and venues, we construct a hierarchical multi-layered document graph with three word relations. (b) For the first $L-1$ convolution steps, we simulate intra-layer propagation within each graph layer. (c) For the L-th convolution, we first average three word relations by mean pooling. (d) We then aggregate auxiliary data across layers to documents. (e) Finally, we use learned topic proportions of documents to reconstruct the corpus.

topics. $\mathbf{d}$ and $\mathbf{a}_d$ are the content and authors of d, respectively. λ_{label} controls label supervision. When labels are not observed, $\lambda_{label} = 0$ for unsupervised learning. λ_{prior} controls prior regularizer.

$q(\cdot) = q(\cdot|\mathcal{D}, \mathcal{A}, \mathcal{V}, \mathcal{X})$ is variational posterior where we omit its conditions to avoid clutter. We also make structured mean-field assumption, $q(\mathbf{Z}_{\mathcal{D}}, \mathbf{Z}_{\mathcal{W}}, \mathbf{Z}_{\mathcal{A}}, \mathbf{Z}_{\mathcal{V}}) = q(\mathbf{Z}_{\mathcal{D}})q(\mathbf{Z}_{\mathcal{W}})q(\mathbf{Z}_{\mathcal{A}})q(\mathbf{Z}_{\mathcal{V}}) = \prod_{d\in\mathcal{D}} q(\mathbf{z}_d) \prod_{w\in\mathcal{W}} q(\mathbf{z}_w) \prod_{a\in\mathcal{A}} q(\mathbf{z}_a) \prod_{v\in\mathcal{V}} q(\mathbf{z}_v)$. The first two rows at Eq. 6 concern data reconstruction, and the next two rows are divergences $\mathcal{R}$ that push variational posteriors to predefined priors as regularization. Eq. 6 is our *objective function* for maximization.

Variational posteriors $q(\cdot)$ are probabilistic encoders parameterized by graph convolutional networks in our model, and log-likelihood, $\log p(\cdot|\cdot)$, in the first two rows are decoders. Below we design the technical details of encoders, decoders, and divergences using the constructed hierarchical multi-layered document graph.

5.1 Graph Convolutional Encoder

We seek a graph convolutional encoder that derives topic proportions for documents preserving both graph structure and corpus semantics. Thus, we propose intra-layer and cross-layer topic propagation for structure modeling and semantic learning, respectively.

5.1.1 ***Intra-Layer Topic Propagation.*** Each graph layer contains one type of vertices and edges. We simulate intra-layer propagation to capture topology of each layer. Due to the heterogeneity of vertices, different types of vertices preserve different feature spaces. To unify heterogeneous vertices, we design a type-specific transformation to project feature spaces of different types to the same low-dimensional space. For a vertex $i \in \mathcal{U}$ with type $o \in O$,

$$\tilde{\mathbf{z}}_i^{(l)} = \mathbf{W}_o^{(l)} \mathbf{z}_i^{(l-1)}. \tag{7}$$

l is the l-th convolutional step. Previous works [11] call it the l-th convolutional layer, but to distinguish it from our multi-layered graph, we call it convolutional step. $\mathbf{z}_i^{(l-1)}$ is the output of previous step, and $\mathbf{z}_i^{(l=0)}$ is the input feature. $\mathbf{W}_o^{(l)}$ is type-specific parameter. Three word sub-layers share the same $\mathbf{W}_o^{(l)}$ due to the same type.

Neighbors of vertex i share semantics with it to different degrees, e.g., some citations discuss similar research, while others are coincidence. We design a type-specific attention within each layer.

$$\alpha_{ij} = \text{softmax}\Big(\text{LeakyReLU}(\mathbf{b}_o^{(l)\top}[\tilde{\mathbf{z}}_i^{(l)} || \tilde{\mathbf{z}}_j^{(l)}])\Big), \quad j \in \mathcal{N}_o(i). \tag{8}$$

$\mathcal{N}_o(i)$ is the set of i's homogeneous neighbors sharing the same type o with vertex i, $[\cdot||\cdot]$ is concatenation operation, and $\mathbf{b}_o^{(l)\top} \in \mathbb{R}^{2k_l}$ is learnable parameter. Finally, we aggregate topics of i's neighbors.

$$\mathbf{z}_i^{(l)} = \tanh\Big(\frac{1}{2}(\tilde{\mathbf{z}}_i^{(l)} + \sum_{j\in\mathcal{N}_o(i)} \alpha_{ij}\tilde{\mathbf{z}}_j^{(l)})\Big). \tag{9}$$

$\mathbf{z}_i^{(l)}$ contains latent topics of both itself and its homogeneous neighbors, and graph structure is captured. We repeat above intra-layer topic propagation until the $(L-1)$-th convolutional step where L is the total number of steps in the encoder network. To summarize,

$$\mathbf{z}_i^{(l)} = f\Big(\mathbf{z}_i^{(l-1)}, \{\mathbf{z}_j^{(l-1)} | j \in \mathcal{N}_o(i)\}\Big), \text{ where } l = 1, 2, ..., L-1. \tag{10}$$

We obtain $\mathbf{z}_{w,ctx}^{(L-1)}$, $\mathbf{z}_{w,syn}^{(L-1)}$, $\mathbf{z}_{w,sem}^{(L-1)}$ for three sub-layers of words; $\mathbf{z}_d^{(L-1)}$, $\mathbf{z}_a^{(L-1)}$, $\mathbf{z}_v^{(L-1)}$ for documents, authors, and venues, respectively. This process is illustrated by Fig. 1(b) where orange arrows denote the direction of intra-layer propagation within each layer.

5.1.2 ***Cross-Layer Topic Propagation.*** We now define the L-th convolutional step. As in previous works [36], as a topic model, our main goal is to use auxiliary information, i.e., authors and venues, to infer topics of documents. We thus focus on document modeling first, after which, we introduce the design of other vertices.

Each document d now has four sets of neighbors, words $\{w_{d,n}\}_{n=1}^{N_d}$, authors $\{a_{d,n}\}_{n=1}^{A_d}$, venue $\{v_d\}$, and homogeneous neighbors $\mathcal{N}_{doc}(d)$ connected by $\mathcal{X}$. Since different sets represent different types, we should distinguish them to preserve corpus heterogeneity. We thus evaluate attention between d and neighbors within each set.

Hierarchical Propagation. We use d's words $\{w_{d,n}\}_{n=1}^{N_d}$ for illustration. Since we model three word relations and obtain $\mathbf{z}_{w,ctx}^{(L-1)}$, $\mathbf{z}_{w,syn}^{(L-1)}$, and $\mathbf{z}_{w,sem}^{(L-1)}$ at Eq. 10 for the same word w, we first unify them by a cross-word-layer mean pooling, illustrated by Fig. 1(c).

$$\mathbf{z}_w^{(L-1)} = \text{mean}(\mathbf{z}_{w,ctx}^{(L-1)}, \mathbf{z}_{w,syn}^{(L-1)}, \mathbf{z}_{w,sem}^{(L-1)}), \tag{11}$$

which is then input to the L-th step. After linear transformation at Eq. 7, we have $\tilde{\mathbf{z}}_d^{(L)}$ and $\tilde{\mathbf{z}}_w^{(L)}$ for document d and word w, respectively. We evaluate attention between document d and its words.

$$\alpha_{d,w} = \text{softmax}\Big(\text{LeakyReLU}(\mathbf{b}^\top [\tilde{\mathbf{z}}_d^{(L)} || \tilde{\mathbf{z}}_w^{(L)}])\Big) \qquad (12)$$

where $w \in \{w_{d,n}\}_{n=1}^{N_d}$, and $\mathbf{b}^\top \in \mathbb{R}^{2k_l}$ is parameter for cross-layer attention. Based on the attention, we aggregate words by

$$\tilde{\mathbf{h}}_w^{(L)} = \sum_w \alpha_{d,w} \tilde{\mathbf{z}}_w^{(L)}, \qquad (13)$$

representing the aggregated topics of d's *whole content*, containing three co-occurrence relations. We use $\mathbf{h}$ to denote the whole neighbors. This process is hierarchical, since each word is first averaged across three word sub-layers, then aggregated with d's other words.

Above we use d's words for illustration. For other types of neighbors, we repeat Eq. 12–13 and obtain $\tilde{\mathbf{h}}_d^{(L)}$, $\tilde{\mathbf{h}}_a^{(L)}$, and $\tilde{\mathbf{h}}_v^{(L)}$, representing d's *whole* homogeneous neighbors, authors, and venues.

Sequence of Authors. When authors are not listed alphabetically, they usually present a sequence of contribution, e.g., academic publications, which reveals the strength of edge connection between these authors and the document. As an author topic model, we aim to incorporate such information and propose a sequence-aware attention. Specifically, when we evaluate attention between document d and its authors $a \in \{a_{d,n}\}_{n=1}^{A_d}$, we extend Eq. 12,

$$\alpha_{d,a} = \text{softmax}\Big(\delta(d,a) \times \text{LeakyReLU}(\mathbf{b}^\top [\tilde{\mathbf{z}}_d^{(L)} || \tilde{\mathbf{z}}_a^{(L)}])\Big). \qquad (14)$$

We add a decay term $\delta(d,a)$, whose value should decrease when the sequence of author a increases. In this paper, we define

$$\delta(d,a) = (1/2)^{s(d,a)-1}. \qquad (15)$$

$s(d,a)$ is the sequence of a in d. $s(d,a) = 1$ if a is the first author. Two authors a_i and a_j with equal contribution have $s(d,a_i) = s(d,a_j)$. Here, the value of 1/2 is chosen, mainly because it performs well on our datasets. Others values are possible, depending on the datasets. Although more complicated attentions are also possible, for simplicity, we design Eq. 15 and leave others as future work.

Reparameterization. Having obtained $\{\tilde{\mathbf{h}}_d^{(L)}, \tilde{\mathbf{h}}_w^{(L)}, \tilde{\mathbf{h}}_a^{(L)}, \tilde{\mathbf{h}}_v^{(L)}\}$ for four graph layers, we propagate them across layers to document d (Fig. 1(d)). η controls the importance of cross-layer propagation.

$$\boldsymbol{\mu}_d = (1-\eta) \times \frac{1}{2}(\tilde{\mathbf{z}}_d^{(L)} + \tilde{\mathbf{h}}_d^{(L)}) + \eta \times \text{mean}(\tilde{\mathbf{h}}_w^{(L)}, \tilde{\mathbf{h}}_a^{(L)}, \tilde{\mathbf{h}}_v^{(L)}) \quad (16)$$

Since we aim to output both mean and covariance from the final convolutional step, we repeat Eq. 11–16 twice and obtain $\boldsymbol{\mu}_d$ and Σ_d for each document d. Assuming isotropic Gaussian with zero mean is the prior, we sample topic proportion $\mathbf{z}_d = \mathbf{z}_d^{(L)} \in \mathbb{R}^K$ by reparameterization trick [10]. For clarity, we omit superscript (L).

$$\mathbf{z}_d = \mathbf{z}_d^{(L)} = \boldsymbol{\mu}_d + (\Sigma_d)^{1/2}\boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \qquad (17)$$

We will analyze the alternatives of Gaussian at Sec. 5.2. Here $\mathbf{z}_d$ is the output of the L-th convolutional step. It contains graph topological structure within each layer by intra-layer propagation, and preserves latent semantics from three relations of words, authors, and venues by cross-layer propagation. To summarize, $\mathbf{z}_d \sim q(\mathbf{z}_d)$ where $q(\mathbf{z}_d)$ is parameterized by our graph convolutional encoder.

We now introduce other vertices. For the final convolution of words, we use Eq. 11 for cross-word-layer mean pooling and obtain $\mathbf{z}_w^{(L-1)}$, which is then input to an intra-layer convolution at Eq. 10.

$$\begin{aligned} \boldsymbol{\mu}_w &= f_\mu\Big(\mathbf{z}_w^{(L-1)}, \{\mathbf{z}_{w'}^{(L-1)} | w' \in \mathcal{N}_{word}(w)\}\Big) \\ \Sigma_w &= f_\Sigma\Big(\mathbf{z}_w^{(L-1)}, \{\mathbf{z}_{w'}^{(L-1)} | w' \in \mathcal{N}_{word}(w)\}\Big). \end{aligned} \qquad (18)$$

Finally, we apply Eq. 17 and obtain $\mathbf{z}_w$ for every word. For authors and venues, we simply repeat intra-layer convolutional step at Eq. 18 and reparameterization at Eq. 17 and output $\mathbf{z}_a$ and $\mathbf{z}_v$.

5.2 Variational Divergence

Having defined graph convolutional encoder as variational posterior $q(\mathbf{z}_i)$, we now turn to the design of the variational divergence term at Eq. 6, which pushes $q(\mathbf{z}_i)$ to a predefined prior $p(\mathbf{z})$ using $\mathcal{R}$ as regularization. Here, we design three modeling alternatives.

5.2.1 ***KL Divergence with Gaussian Prior.*** Following VGAE [12], the first design is KL divergence as $\mathcal{R}$ and isotropic Gaussian with zero mean as prior $p(\mathbf{z})$. Above reparameterization at Eq. 17 follows this Gaussian prior. The corresponding KL divergence is

$$\text{KL}[q(\mathbf{z}_i)||p(\mathbf{z})] = \frac{1}{2}(\text{tr}(\Sigma_i) + \boldsymbol{\mu}_i^\top \boldsymbol{\mu}_i - \log|\Sigma_i| - K). \qquad (19)$$

Vertex $i \in \mathcal{U}$. $q(\mathbf{z}_i)$ is our graph encoder, which outputs $\boldsymbol{\mu}_i$ and Σ_i as Gaussian variational posterior. $\text{tr}(\cdot)$ is the trace of a matrix.

5.2.2 ***KL Divergence with Dirichlet Prior.*** Inspired by the success of Dirichlet prior in LDA [2], which improves topic quality, we analyze Dirichlet prior as an alternative of Gaussian. We follow [29] and evaluate Dirichlet posterior $q(\mathbf{z}_i)$ by Laplace approximation.

$$q(\mathbf{z}_i) = \text{softmax}(\boldsymbol{\mu}_i + \Sigma_i^{1/2}\boldsymbol{\epsilon}), \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \qquad (20)$$

Having defined posterior, we approximate predefined Dirichlet prior $p(\mathbf{z}) = \text{Dir}(\alpha)$. We calculate its mean $\boldsymbol{\mu}$ and covariance Σ by

$$\mu_k = \log \alpha_k - \frac{1}{K}\sum_{k'} \log \alpha_{k'}, \quad \Sigma_{kk} = \frac{1}{\alpha_k}(1-\frac{2}{K}) + \frac{1}{K^2}\sum_{k'}\frac{1}{\alpha_{k'}} \quad (21)$$

where Σ is a diagonal matrix. After obtaining $\boldsymbol{\mu}$ and Σ, we use Eq. 20 to get approximated Dirichlet prior $p(\mathbf{z})$. KL divergence is

$$\text{KL}[q(\mathbf{z}_i)||p(\mathbf{z})] = \tfrac{1}{2}\Big(\text{tr}(\Sigma^{-1}\Sigma_i) + (\boldsymbol{\mu} - \boldsymbol{\mu}_i)^\top \Sigma^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_i) + \log\tfrac{|\Sigma|}{|\Sigma_i|} - K\Big). \qquad (22)$$

5.2.3 ***Wasserstein Distance with Gaussian Prior.*** Variational divergence consists of three components, i.e., variational posterior $q(\mathbf{z}_i)$ defined by our graph convolutional encoder, predefined prior $p(\mathbf{z})$ investigated above, and divergence metric $\mathcal{R}$. One drawback of KL is that it is not symmetric and does not obey triangle inequality, which influences the measure of distributions in Euclidean space. We thus analyze $\mathcal{R}$ and seek an alternative of KL. Inspired by WLDA [25], which uses Wasserstein distance in the word space and achieves improvement, we analyze Wasserstein distance in the topic space. Convolutional encoder outputs $\boldsymbol{\mu}_i$ and Σ_i as Gaussian variational posterior. We measure its distance with Gaussian prior.

THEOREM 5.1. *Let $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and $q(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$ be two Gaussian distributions. Their 2-Wasserstein distance is [37]*

$$W_2[p(\boldsymbol{z}), q(\boldsymbol{z}_i)] = ||\boldsymbol{\mu} - \boldsymbol{\mu}_i||_2^2 + tr(\Sigma + \Sigma_i - 2(\Sigma^{\frac{1}{2}}\Sigma_i\Sigma^{\frac{1}{2}})^{\frac{1}{2}}). \qquad (23)$$

Wasserstein distance between two Gaussians has an analytical solution. Specifically, in our model the covariance of Gaussian prior and variational posterior is diagonal, $\Sigma = \text{diag}(\sigma^2)$ and $\Sigma_i = \text{diag}(\sigma_i^2)$, Eq. 23 can be simplified as a symmetric form

$$W_2[p(\mathbf{z}), q(\mathbf{z}_i)] = ||\boldsymbol{\mu} - \boldsymbol{\mu}_i||_2^2 + ||\boldsymbol{\sigma}^2 - \boldsymbol{\sigma}_i^2||_2^2. \tag{24}$$

We will examine the effect of these three modeling alternatives.

5.3 Probabilistic Decoder

We now design a decoder to generate the observed data, which is the log-likelihood reconstruction $\log p(\cdot|\cdot)$ at objective Eq. 6.

Specifically, we use textual content generation for illustration. For a document $d \in \mathcal{D}$, $\log p(\mathbf{d}|\mathbf{z}_d, \mathbf{Z}_{\mathcal{W}})$ at Eq. 6 is the log-likelihood of content generation where $\mathbf{z}_d = \mathbf{z}_d^{(L)}$ and $\mathbf{Z}_{\mathcal{W}} = [\mathbf{z}_{w_1}^{(L)}; \mathbf{z}_{w_2}^{(L)}; ...]^\top \in \mathbb{R}^{|\mathcal{W}| \times K}$ are the outputs of the graph convolutional encoder. We define $\log p(\mathbf{d}|\mathbf{z}_d, \mathbf{Z}_{\mathcal{W}}) = \sum_{w \in \mathbf{d}} \log[\phi(\mathbf{Z}_{\mathcal{W}} \mathbf{z}_d)^{d_w} (1 - \phi(\mathbf{Z}_{\mathcal{W}} \mathbf{z}_d))^{1-d_w}]$, where $d_w = 1$ if word w appears in document d, otherwise $d_w = 0$. $\phi(x) = \frac{1}{1+\exp(-x)}$ is sigmoid. We use inner product of document's and words' topic proportions to predict each word. However, above equation inefficiently requires summation over the entire vocabulary. Empirically, we use negative sampling [24] to replace it.

$$\sum_{w:d_w=1} [\log \phi(\mathbf{z}_d^\top \mathbf{z}_w) + \sum_{m=1}^{M} \mathbb{E}_{w' \sim p_n(w)} \log \phi(-\mathbf{z}_d^\top \mathbf{z}_{w'})] \tag{25}$$

M is the number of negative samples, $p_n(w)$ is a noise distribution. Above we use content generation for illustration. For authors, venues, and connected documents, the reconstruction terms (Eq. 25) are similarly defined by replacing $\mathbf{z}_w$ with $\mathbf{z}_a$, $\mathbf{z}_v$, and $\mathbf{z}_d$, respectively. This decoding process is shown by Fig. 1(e) by red arrows.

If document d's label exists, we define label generation by

$$\hat{\mathbf{y}}_d = \text{softmax}(f_{\text{MLP}}(\mathbf{z}_d)), \quad \log p(\mathbf{y}_d|\mathbf{z}_d) = \sum_n y_{d,n} \log \hat{y}_{d,n}. \tag{26}$$

$f_{\text{MLP}}(\cdot)$ is a multi-layer perceptron, $\mathbf{y}_d$ is a one-hot label encoding.

Up to now, we have elaborated all three modeling components. Graph convolutional encoder simulates intra- and cross-layer topic propagation on a hierarchical multi-layered document graph to capture graph structure and latent semantics. Variational divergence analyzes predefined prior and divergence metric. Decoder generates the observations with both supervised and unsupervised version. We optimize objective function Eq. 6 until convergence. Algo. 1 at Appendix summarizes the training process of our model.

6 EXPERIMENTS

The main objective is to evaluate the quality of documents' topics learned from a corpus with auxiliary authorship and venues.

Datasets. We use six datasets at Table 2. Cora [22] is a corpus of papers with abstract as content and citations as doc-doc edges. Each paper has a sequence of authors. We extracted two independent datasets, Machine Learning (ML) and Programming Language (PL). Besides, we used two more datasets, HEP-TH [14] and Aminer [30] as Physics and CS paper corpus, both with authors and venues. COVID is a Coronavirus news corpus[1]. Each article has an editor and published on a platform. Since no doc-doc edges are observed, we generate κNN edges using Bag-of-Words similarity ($\kappa = 5$). Web [13] is a Web page hyperlink network. Each page is a news article and associated with an author. See Appendix A.2 for more details.

Table 2: Dataset statistics.

Name	#Documents	#Authors	#Venues	#Doc-Doc Edges	Vocabulary	#Labels
ML	2,947	2,814	N.A.	8,146	5,814	7
PL	2,449	2,778	N.A.	7,274	5,066	9
COVID	1,500	880	169	5,706	5,083	5
HEP-TH	20,151	10,432	343	234,193	5,001	N.A.
Aminer	114,741	143,534	50	265,345	10,018	10
Web	445,657	36,405	N.A.	565,502	10,015	N.A.

Baselines. We consider 5 categories of baselines. *i)* **Topic models for plain text**, ProdLDA [29], WLDA [25], NSTM [44], and DVAE [3]. ProdLDA and DVAE use Dirichlet as predefined prior. WLDA uses Wasserstein distance in the word space. These *unsupervised* models are not proposed for author or venue modeling. To allow them to model authors and venues, we consider each author and venue as a document, and the content is the aggregation of associated documents. *ii)* **Author topic models** deal with corpus with authors, we compare to ATM [28] where topics of a document are the average of its authors'. *iii)* **Topic models for document graphs**, RTM [4], Adjacent-Encoder [40], and LANTM [33]. They construct a document graph and learn topic proportions in an *unsupervised* way. We extend them to consider authorship by running on our constructed multi-layered graph. *iv)* **Text classification models** learn text embeddings with *label supervision* for classification. We mainly compare to graph models, TextGCN [39], HyperGAT [6], TVGAE [36]. TextGCN and HyperGAT are not topic models, since text embeddings are not interpretable topics. TVGAE integrates topic model into VGAE. We allow them to model authors and venues by converting authors and venues as documents. *v)* **Graph embedding models** are not topic models, either. For completeness, we consider HAN [32] as *supervised* and VGAE [12] as *unsupervised* method, both with authors and venues.

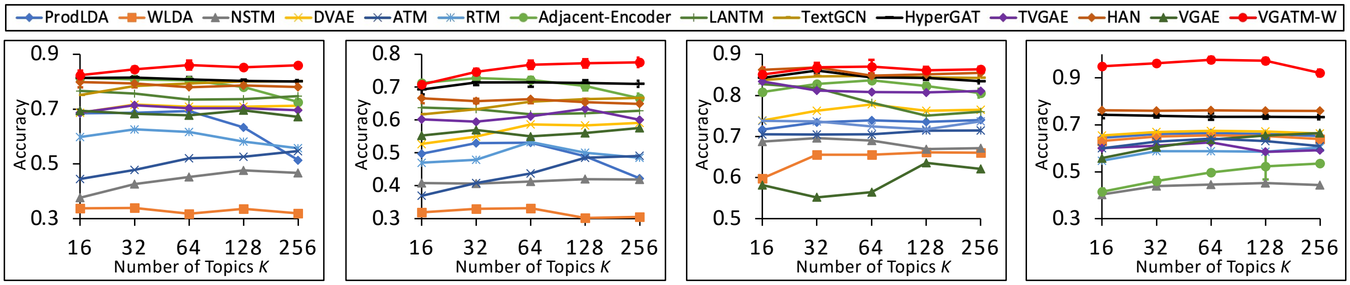
We set two convolutional steps for our model. We present three variants, VGATM-G, VGATM-D, and VGATM-W, for Gaussian prior, Dirichlet prior, and Wasserstein distance, respectively. $\lambda_{prior} = 0.01$, $\eta = 0.1$, and $M = 5$. For our supervised version, $\lambda_{label} = 1$. For VGATM-D, $\alpha = 1$ for Dirichlet prior. For HAN, the combination of metapaths {DAD, DWD, DVD, DD} performs the best. Each result is obtained by 5 independent runs. We report mean and std.dev. Code and datasets are available at https://github.com/cezhang01/vgatm.

6.1 Quantitative Evaluation

6.1.1 ***Document Classification.*** Following LDA [2], to evaluate topic quality, we rely on document classification. Given a corpus, we split 80% documents for training, among which 10% are for validation. We also observe authors, venues, graph edges, and labels associated with training documents. During test, we infer topics of test documents and classify them. Since we have both supervised and unsupervised version, we conduct two classification tasks.

Supervised Training. Labels are involved for supervised training. We compare to all baselines. Supervised baselines output predicted labels for documents, which are then compared with ground-truth labels. For completeness, we also compare to unsupervised baselines, which output topic proportions without label prediction.

[1] https://aylien.com/blog/free-coronavirus-news-dataset



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

Figure 2: Supervised document classification when varying the number of topics K from 16 to 256.

Table 3: Unsupervised classification (in percentage) at $K = 64$.

Model	ML	PL	COVID	Aminer
ProdLDA	69.3±0.7	53.1±2.5	73.9±1.6	64.0±0.2
WLDA	31.8±3.5	33.2±1.8	65.6±2.5	65.5±0.2
NSTM	45.2±2.6	41.3±3.2	69.0±2.1	44.6±0.3
DVAE	70.8±1.3	58.7±1.5	77.8±2.1	67.4±0.3
ATM	52.0±1.3	43.8±3.0	72.4±1.7	64.1±0.8
RTM	61.6±2.4	53.3±1.4	70.5±3.2	58.8±0.5
Adjacent-Encoder	80.5±0.6	72.2±0.9	83.7±1.0	49.6±0.3
LANTM	73.5±1.6	61.8±0.9	78.2±1.6	N.A.
VGAE	67.7±1.9	55.0±2.3	56.4±4.6	63.6±0.6
VGATM-G	81.5±0.4	73.7±0.5	83.2±1.1	98.0±0.1
VGATM-D	82.5±0.7	73.1±0.7	83.6±0.6	**98.9±0.2**
VGATM-W	**84.4±0.3**	**74.8±1.2**	**84.7±1.3**	97.7±0.4

We follow [40] and train an external kNN classifier ($k = 5$) using the output topics of training documents and predict labels of test documents. Fig. 2 shows classification accuracy with different number of topics. We exclude LANTM and TextGCN on large dataset Aminer, since they cannot run even on a machine with 256GB memory.

Unsupervised Training. We set $\lambda_{label} = 0$ and do not observe labels for training. For a fair comparison, we compare against unsupervised baselines only. We use kNN as external classifier for both our models and baselines. Table 3 shows the accuracy at 64 topics.

Analysis. For both classification tasks, the best baselines are Adjacent-Encoder, LANTM, and HAN, which model document graph but ignore three word relations. In contrast, we consider contextual, syntactic, and semantic relations, and improve the result. VGATM-W is the best one among our variants at Table 3, which verifies that Wasserstein is a promising alternative of KL. Dirichlet prior performs better than Gaussian. As verified by previous work [3], Dirichlet encourages topics to be sparser than Gaussian and achieves a lower reconstruction error, thus improving topic quality.

6.1.2 *Link Prediction.*

Edges reveal semantic similarity between documents. As in RTM [4], we conduct link prediction to evaluate topic quality. As in [40], the first task is doc-doc link prediction. Besides, as an author topic model, we also predict authors given a document, i.e., doc-author link prediction in our document graph.

Doc-Doc Link Prediction. During training, we observe 80% training documents and links within them. During test, we predict links within 20% test documents. As in [40], the probability of a link is $p(x_{d_i,d_j}|\mathbf{z}_{d_i}, \mathbf{z}_{d_j}) \propto \exp(-||\mathbf{z}_{d_i} - \mathbf{z}_{d_j}||_2^2)$. We compare the predicted probability against the ground-truth adjacency by AUC [33]. Table 4(left) shows the results. LANTM and TextGCN cannot run on large datasets and do not have results. Supervised models (TextGCN, HyperGAT, TVGAE, and HAN) require labels for training, thus cannot run on HEP-TH and Web with no labels.

Doc-Author Link Prediction. We then predict authors given a document. For authors with at least three documents, we randomly remove one document as the test doc-author links. We input the remaining corpus to train the model. After convergence, we predict the held-out links. Table 4 (right) summarizes the results.

Analysis. For both scenarios, our models predict links more accurately than baselines. Compared to models with plain text, we show the advantage of constructing document graph using auxiliary authors and venues. Compared to models with graph structure, we verify the benefit of modeling three word co-occurrence relations.

6.2 Topic Analysis

6.2.1 *Topic Coherence.*

One advantage of topic models is semantic interpretability: each topic is interpreted by its key words. $\mathbf{Z}_{\mathcal{W}} \in \mathbb{R}^{|\mathcal{W}| \times K}$ is topic-word distribution. Each column is the distribution of a topic over the words, and the highest values on that column are the key words of that topic. As in ProdLDA [29], we evaluate the coherence of key words by an external corpus, Google Web 1T 5-gram Version 1 [7], with NPMI as metric. Table 5 (left) shows the results. We exclude TextGCN, HyperGAT, HAN, VGAE, since they are not topic models. TVGAE is a supervised topic model, thus cannot run on HEP-TH and Web with no labels. Our models outperform baselines except one case: NSTM learns more coherent topics on ML, possibly because it models pretrained word embeddings. VGATM-D is better than VGATM-G, since Dirichlet prior achieves low reconstruction error, producing more coherent topics.

6.2.2 *Perplexity.*

Following LDA [2], we evaluate perplexity to analyze topic quality. We evaluate perplexity for 20% test documents. Perplexity, $\exp\{-\frac{\log p(\mathcal{D}_{test})}{\sum_{d \in \mathcal{D}_{test}} N_d}\}$, is exponential and varies much w.r.t. its power, we thus present its power $-\frac{\log p(\mathcal{D}_{test})}{\sum_{d \in \mathcal{D}_{test}} N_d}$ for clarity (smaller is better). Table 5 (right) shows that our models consistently outperform baselines. Benefiting from document graph with authors and venues, Adjacent-Encoder presents the lowest perplexity among baselines. Compared to it, our models further consider three word relations, improving ours over Adjacent-Encoder.

6.2.3 *Interpretability.*

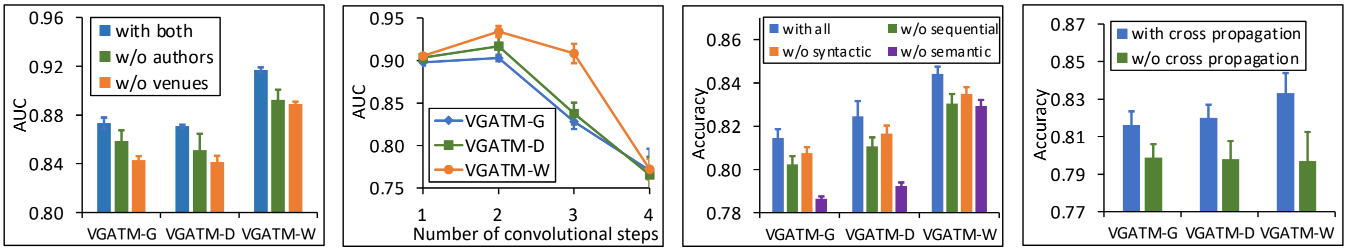
To understand what topics our models capture, we randomly select two topics for each variant and present

Table 4: Link prediction AUC (in percentage) with doc-doc link prediction (left) and doc-author link prediction (right) at $K = 64$.

Category	Model	Doc-Doc Link Prediction						Doc-Author Link Prediction					
		ML	PL	COVID	HEP-TH	Aminer	Web	ML	PL	COVID	HEP-TH	Aminer	Web
Models for plain text	ProdLDA	81.8±0.8	74.9±0.6	75.5±1.0	64.2±2.2	80.2±0.4	82.4±0.0	65.3±0.0	67.1±0.0	26.8±1.5	45.0±1.5	54.3±0.2	60.5±0.0
	WLDA	52.4±0.8	54.7±1.0	67.1±1.3	62.8±0.4	79.7±0.7	79.3±0.5	31.9±0.6	31.1±0.4	33.0±1.3	33.0±0.3	47.4±0.5	35.6±1.2
	NSTM	63.2±1.7	62.4±0.7	66.3±1.4	58.3±0.3	58.8±0.6	67.0±0.8	51.2±1.3	49.9±0.5	44.9±2.8	44.1±0.3	47.2±0.2	59.5±0.0
	DVAE	79.9±0.8	73.1±0.4	73.4±0.2	82.0±0.1	89.8±0.3	88.3±0.0	64.8±0.3	62.9±0.8	49.4±0.7	66.7±0.3	66.3±0.2	71.7±0.0
Author topic models	ATM	71.1±1.6	69.2±1.2	61.0±0.2	66.8±0.3	64.4±0.4	87.6±0.0	40.6±2.5	37.7±1.6	29.6±4.0	57.7±0.6	70.1±0.5	59.6±2.1
Models with document graph	RTM	71.0±1.0	68.1±0.5	70.5±0.3	69.7±0.8	77.5±0.7	78.4±0.1	32.1±0.4	32.7±0.1	32.2±0.4	30.2±0.0	25.8±0.1	34.9±0.1
	Adjacent-Encoder	84.7±0.9	84.9±1.9	94.7±0.4	75.0±0.6	71.8±0.7	73.2±0.0	90.2±0.6	89.7±0.2	73.6±1.2	75.3±0.7	37.9±0.0	36.2±0.0
	LANTM	80.6±1.2	75.4±0.7	84.9±1.1	86.1±0.3	N.A.	N.A.	86.1±0.9	87.8±0.8	71.0±1.5	85.7±0.3	N.A.	N.A.
Text classification models (they are *supervised* and cannot run on HEP-TH and Web with no observed labels)	TextGCN	81.3±0.3	75.4±0.4	81.1±0.1	N.A.	N.A.	N.A.	56.8±0.7	50.4±1.6	47.7±5.2	N.A.	N.A.	N.A.
	HyperGAT	83.1±0.5	79.7±0.5	87.1±0.3	N.A.	90.0±0.0	N.A.	50.0±0.8	49.6±0.7	61.8±3.1	N.A.	49.1±0.2	N.A.
	TVGAE	79.1±0.7	74.7±1.0	88.2±1.0	N.A.	85.3±0.6	N.A.	65.0±0.9	65.4±0.9	72.8±1.5	N.A.	70.6±0.7	N.A.
Graph embedding models (HAN is *supervised*, cannot run without labels)	HAN	77.0±0.7	73.1±0.4	84.7±1.0	N.A.	93.2±0.1	N.A.	73.0±1.4	72.2±2.2	79.2±1.1	N.A.	71.3±1.1	N.A.
	VGAE	72.5±0.5	80.4±0.2	84.1±2.8	72.7±1.7	91.9±0.6	87.4±0.2	82.3±2.3	86.3±1.2	63.8±3.2	77.7±3.3	64.9±0.9	73.8±1.9
Our proposed models	VGATM-G	91.3±0.7	91.1±0.5	91.1±0.5	86.3±0.5	94.5±0.4	93.0±0.1	92.0±0.3	93.1±0.1	73.7±2.0	90.0±0.3	72.9±0.9	76.1±1.0
	VGATM-D	91.7±1.2	90.6±0.2	91.3±0.3	87.1±0.1	94.4±0.4	93.0±0.2	92.3±0.3	93.2±0.4	74.9±0.6	90.3±0.3	74.0±1.0	76.2±0.4
	VGATM-W	**93.4±0.4**	**92.1±0.2**	**95.4±0.3**	**91.7±0.2**	**95.5±1.0**	**93.5±0.4**	**93.0±0.3**	**93.8±0.5**	**79.5±1.2**	**91.2±0.3**	**74.1±0.3**	**77.3±0.0**

Table 5: Topic coherence NPMI (left) and perplexity (right) at $K = 64$.

Category	Model	Topic Coherence NPMI						Perplexity					
		ML	PL	COVID	HEP-TH	Aminer	Web	ML	PL	COVID	HEP-TH	Aminer	Web
Models for plain text	ProdLDA	10.0±0.7	9.4±0.5	12.0±0.7	10.3±0.6	9.3±0.5	21.2±0.2	7.19±0.00	7.21±0.00	7.82±0.00	7.72±0.00	8.18±0.00	8.34±0.00
	WLDA	9.7±0.2	11.6±0.1	12.5±0.5	13.7±0.4	17.9±0.5	23.9±0.8	18.90±0.73	19.57±0.30	28.56±1.09	44.31±0.18	44.67±0.10	45.22±0.00
	NSTM	**16.0±1.0**	18.6±0.6	22.0±0.6	18.2±0.5	15.5±0.3	24.0±0.3	8.46±0.00	8.34±0.00	8.38±0.00	8.39±0.00	9.00±0.00	8.93±0.00
	DVAE	14.7±0.0	15.2±0.1	15.8±0.1	14.8±0.1	15.5±0.1	17.6±0.2	17.74±0.14	18.96±0.08	17.16±0.26	23.67±0.11	40.50±0.04	43.32±0.00
Author topic models	ATM	10.2±0.4	12.0±0.5	9.8±0.2	10.2±0.3	15.0±0.2	23.2±0.7	6.63±0.01	6.45±0.01	7.33±0.04	7.05±0.00	7.65±0.01	7.21±0.00
Models with document graph (LANTM cannot run on large dataset Aminer and Web even on 256GB machine)	RTM	7.3±0.2	8.9±0.5	16.2±0.5	6.6±0.3	10.8±0.3	20.9±0.4	8.07±0.01	7.93±0.01	8.98±0.04	8.04±0.00	8.89±0.01	10.28±0.19
	Adjacent-Encoder	12.4±0.9	12.5±0.7	13.8±0.4	13.4±0.4	11.4±0.2	15.2±0.1	7.41±0.01	7.34±0.13	6.96±0.00	7.45±0.19	8.71±0.02	8.26±0.01
	LANTM	9.9±1.2	9.8±0.7	8.6±0.3	10.4±1.5	N.A.	N.A.	8.63±0.00	8.48±0.00	8.48±0.00	8.50±0.00	N.A.	N.A.
Text classification (cannot run with no labels)	TVGAE	3.3±0.5	3.8±0.5	5.2±0.5	N.A.	2.6±0.3	N.A.	10.53±0.27	10.13±0.53	11.30±0.47	N.A.	10.24±0.17	N.A.
Our proposed models	VGATM-G	13.2±0.7	19.6±1.9	19.7±0.9	15.5±1.0	21.5±0.7	19.6±0.6	5.50±0.24	5.64±0.26	6.95±0.09	5.06±0.05	5.78±0.13	**5.29±0.14**
	VGATM-D	13.0±0.8	19.3±2.8	**22.9±1.8**	15.8±0.8	20.9±0.3	**26.4±2.8**	5.36±0.12	5.62±0.24	6.80±0.16	5.04±0.09	5.94±0.24	6.40±0.33
	VGATM-W	13.6±1.1	**20.5±1.0**	19.4±1.8	**19.0±0.0**	**21.7±1.1**	23.7±1.7	**5.23±0.15**	**5.13±0.30**	**6.55±0.20**	**4.94±0.06**	**5.75±0.28**	5.60±0.41



(a) Effect of authors and venues (b) Number of convolutional steps (c) Three word co-occurrence patterns (d) Effect of cross-layer propagation

Figure 3: Ablation analysis of our models.

Table 6: Top-5 words of 2 randomly selected topics of VGATM.

Model	Topic	Key words
VGATM-G	1	hospital, nurse, children, died, clinic
	2	manufacturing, import, affected, slowdown, agricultural
VGATM-D	1	employee, employees, retirees, worker, insurance
	2	rugby, club, illness, match, championship
VGATM-W	1	classwork, loved, classmates, no-one, at-home
	2	cases, patients, disease, diseases, deaths

top-5 words on COVID at Table 6. VGATM-G captures *children's health* and *manufacture depression*. VGATM-D reveals *retirement* and *sports*. VGATM-W shows *studying at home* and *confirmed cases*.

6.3 Model Analysis

6.3.1 ***Effect of Authors and Venues.*** We test the effect of authors and venues. We respectively remove authors and venues, and use the remaining corpus for training. Fig. 3(a) presents doc-doc link prediction results on HEP-TH. Our models with both information perform the best, showing the advantage of authors and venues. We conclude that venues are more informative on HEP-TH, since the result drops more when removing venues than removing authors.

6.3.2 ***Number of Convolutional Steps.*** We analyze the performance of different convolutional steps L at Fig. 3(b), doc-doc link prediction on ML dataset. When $L = 1$, we cannot fully capture high-order neighbors, leading to inferior results. When $L = 2$, we observe an increasing trend. However, an overly high value of L hurts the result, since further neighbors with noise are modeled.

6.3.3 ***Three Word Co-occurrence Relations.*** Here we test the effectiveness of three word relations by removing each one from the complete models. Fig. 3(c) shows classification accuracy on ML. Models with all three relations outperform other versions, verifying that we indeed capture every word relation to improve topic modeling. Semantic relation plays the most important role, since disregarding it leads to the worst accuracy. Syntactic relation is less informative, since removing it does not hurt the result much.

6.3.4 *Effect of Cross-Layer Topic Propagation.*

Cross-layer topic propagation integrates auxiliary information into topic proportions of documents. To test its usefulness, we remove it by setting $\eta = 0$ at Eq. 16 and maintain intra-layer propagation only. Fig. 3(d) summarizes classification accuracy on ML dataset. We conclude that cross-layer topic propagation allows topics of documents to better capture auxiliary information and improves topic quality.

7 CONCLUSION

We propose VGATM, working under supervised and unsupervised settings. To model authors, venues, and three word relations, we design a hierarchical multi-layered graph and three alternatives of divergence. Experiments verify the effectiveness of VGATM.

REFERENCES

[1] Haoli Bai, Zhuangbin Chen, Michael R Lyu, Irwin King, and Zenglin Xu. 2018. Neural relational topic models for scientific article analysis. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 27–36.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3 (2003), 993–1022.

[3] Sophie Burkhardt and Stefan Kramer. 2019. Decoupling Sparsity and Smoothness in the Dirichlet Variational Autoencoder Topic Model. *J. Mach. Learn. Res.* 20, 131 (2019), 1–27.

[4] Jonathan Chang and David Blei. 2009. Relational topic models for document networks. In *Artificial intelligence and statistics*. PMLR, 81–88.

[5] Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. 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)*. 795–804.

[6] Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be More with Less: Hypergraph Attention Networks for Inductive Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 4927–4936.

[7] Stefan Evert. 2010. Google web 1t 5-grams made easy (but not for the computer). In *Proceedings of the NAACL HLT 2010 Sixth Web as Corpus Workshop*. 32–40.

[8] Aditya Grover, Aaron Zweig, and Stefano Ermon. 2019. Graphite: Iterative generative modeling of graphs. In *International conference on machine learning*. PMLR, 2434–2444.

[9] Yoon Kim. 2014. Convolutional neural networks for sentence classification. EMNLP.

[10] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[11] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[12] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[13] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 497–506.

[14] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 177–187.

[15] Jia Li, Jianwei Yu, Jiajin Li, Honglei Zhang, Kangfei Zhao, Yu Rong, Hong Cheng, and Junzhou Huang. 2020. Dirichlet Graph Variational Autoencoder. *Advances in Neural Information Processing Systems* 33 (2020).

[16] Kar Wai Lim and Wray Buntine. 2015. Bibliographic analysis with the citation network topic model. In *Asian conference on machine learning*. PMLR, 142–158.

[17] Hu Linmei, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. 2019. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 4821–4830.

[18] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 2873–2879.

[19] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L Gaunt. 2018. Constrained graph variational autoencoders for molecule design. In *Advances in Neural Information Processing Systems*. 7806–7815.

[20] Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. 2020. Tensor graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8409–8416.

[21] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 55–60.

[22] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval* 3, 2 (2000), 127–163.

[23] Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International conference on machine learning*. PMLR, 1727–1736.

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[25] Feng Nan, Ran Ding, Ramesh Nallapati, and Bing Xiang. 2019. Topic Modeling with Wasserstein Autoencoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 6345–6381.

[26] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. 2018. Adversarially regularized graph autoencoder for graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2609–2615.

[27] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[28] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. 487–494.

[29] Akash Srivastava and Charles Sutton. 2017. Autoencoding Variational Inference for Topic Models. In *5th International Conference on Learning Representations*.

[30] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 990–998.

[31] Yuancheng Tu, Nikhil Johri, Dan Roth, and Julia Hockenmaier. 2010. Citation author topic model in expert search. In *Coling 2010: Posters*. 1265–1273.

[32] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*. 2022–2032.

[33] Yiming Wang, Ximing Li, and Jihong Ouyang. 2021. Layer-assisted neural topic modeling over document networks. In *International Joint Conference on Artificial Intelligence*. 3148–3154.

[34] Zhengjue Wang, Chaojie Wang, Hao Zhang, Zhibin Duan, Mingyuan Zhou, and Bo Chen. 2020. Learning dynamic hierarchical topic graph with graph convolutional network for document classification. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3959–3969.

[35] Qianqian Xie, Jimin Huang, Pan Du, Min Peng, and Jian-Yun Nie. 2021. Graph topic neural network for document representation. In *Proceedings of the Web Conference 2021*. 3055–3065.

[36] Qianqian Xie, Jimin Huang, Pan Du, Min Peng, and Jian-Yun Nie. 2021. Inductive Topic Variational Graph Auto-Encoder for Text Classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4218–4227.

[37] Hongteng Xu, Dixin Luo, Ricardo Henao, Svati Shah, and Lawrence Carin. 2020. Learning autoencoders with relational regularization. In *International Conference on Machine Learning*. PMLR, 10576–10586.

[38] Liang Yang, Fan Wu, Junhua Gu, Chuan Wang, Xiaochun Cao, Di Jin, and Yuanfang Guo. 2020. Graph attention topic modeling network. In *Proceedings of The Web Conference 2020*. 144–154.

[39] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 7370–7377.

[40] Ce Zhang and Hady W Lauw. 2020. Topic modeling on document networks with adjacent-encoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6737–6745.

[41] Delvin Ce Zhang and Hady W Lauw. 2021. Representation Learning on Multi-layered Heterogeneous Network. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 399–416.

[42] Delvin Ce Zhang and Hady W Lauw. 2021. Semi-supervised semantic visualization for networked documents. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 762–778.

[43] Hao Zhang, Bo Chen, Dandan Guo, and Mingyuan Zhou. 2018. WHAI: Weibull Hybrid Autoencoding Inference for Deep Topic Modeling. In *International Conference on Learning Representations*.

[44] He Zhao, Dinh Phung, Viet Huynh, Trung Le, and Wray Buntine. 2020. Neural Topic Model via Optimal Transport. In *ICLR*.

[45] Qile Zhu, Zheng Feng, and Xiaolin Li. 2018. GraphBTM: Graph enhanced autoencoded variational inference for biterm topic model. In *Proceedings of the 2018 conference on empirical methods in natural language processing*. 4663–4672.

Algorithm 1 Training Process of VGATM

```
Input: Corpus $C$ with documents $\mathcal{D}$, authors $\mathcal{A}$, venues $\mathcal{V}$, and doc-doc edge connections $\mathcal{X}$, number of convolutional steps $L$, number of topics $K$, and number of negative samples $M$.
Output: Topic proportions $\mathbf{Z}_{\mathcal{D}}, \mathbf{Z}_{\mathcal{W}}, \mathbf{Z}_{\mathcal{A}}, \mathbf{Z}_{\mathcal{V}}$.
Initialize model parameters $\mathbf{W}_o^{(l)}$, $\mathbf{b}_o^{(l)}$, and $\mathbf{b}$, for $l = 1, 2, ..., L$.
while not converged do
    //intra-layer topic propagation
    for $l = 1, 2, ..., L-1$ do
        Simulate intra-layer topic propagation by Eq. 7–10.
    end for
    //cross-layer topic propagation
    Evaluate topic proportion of document $d$'s whole content, homogeneous neighbors, authors, and venues by Eq. 11–13.
    Propagate $d$'s auxiliary data across layers to $d$ by Eq. 16.
    Reparameterization by Eq. 17–18 as the output of encoder.
    //variational divergence
    Evaluate variational divergence Eq. 19–24.
    //probabilistic decoder
    Reconstruct corpus Eq. 25–26 using the output of encoder.
    //optimization
    Maximize objective function Eq. 6.
end while
```

A REPRODUCIBILITY SUPPLEMENT

A.1 Pseudo-Code of Training Process

We summarize the training process of our model at Algo. 1.

A.2 Dataset Preprocessing

In this section, we introduce the details of dataset preprocessing.

- **ML** and **PL** are constructed from Cora. We maintained documents associated with a sequence of authors, resulting in 2,947 and 2,449 documents, respectively. We were not able to obtain the venue information of these documents. For both datasets, after removing stop words and punctuations, we maintained the most frequent 5,000 words as vocabulary.
- **COVID** is a publicly available Coronavirus news corpus[2]. It is a collection of news articles related to coronavirus from multiple publishers since the outbreak in late 2019. Each news article is associated with one category, which we treat as label. We selected 5 categories, namely, *economy, business, and finance*, *education*, *health*, *labour*, and *sports*. For each category, we randomly selected 300 news articles, generating a corpus of 1,500 documents and 5 labels. Each article has an editor and is published on a news module. In total, we obtained 880 editors as authors and 169 modules as venues. After removing stop words and punctuations, we maintained the most frequent 5,000 words as vocabulary. Since no appropriate doc-doc edges are given, following [33], we generated κNN edges on documents' Bag-of-Words similarity. We did not observe much difference from $\kappa = 5$ to $\kappa = 15$, for efficiency, we set $\kappa = 5$, resulting in 5,706 edges in total.

Table 7: Categories and venues of Aminer dataset

Category	Venues
Computational Linguistics	ACL, EMNLP, NAACL, COLING, EACL
Databases and Information Systems	SIGMOD, VLDE, ICDE, CIKM, IPM
Data Mining and Analysis	KDD, WWW, ICDM, TKDE, SIGIR
Computer Vision and Pattern Recognition	CVPR, ICCV, ECCV, TPAMI, TIP
Artificial Intelligence	NeurIPS, ICML, AAAI, IJCAI, JMLR
Computer Graphics	TOG, TVCG, SIGGRAPH, CGA, TVS
Theoretical Computer Science	STOC, SODA, FOCS, JOC, JACM
Software Systems	ICSE, ASE, FSE, TSE, PLDI
Computer Networks and Wireless Communication	SIGCOMM, INFOCOM, TWC, CM, JNCA
Computing Systems	TPDS, ISCA, TJSC, ICDCS, ATC

- **HEP-TH** is a Physics paper corpus with abstract as document content and citations as doc-doc edges. We extracted documents with a sequence of authors and publication venues, resulting in 20,151 documents, 10,432 authors, and 343 venues. Similarly, we removed stop words and punctuations, and maintained the most frequent 5,000 words as vocabulary. The original dataset does not contain labels of documents.
- **Aminer** is an academic paper corpus[3] where each paper is associated with a sequence of authors and published on a journal or a conference. We used *ACM-Citation-network V8* as raw dataset. Since we did not discover any explicit labels of documents, we labeled documents based on their publication venues. Specifically, we used Google Scholar Metrics[4] as ground-truth categories. We selected 10 computer science categories, and for each category, we selected 5 conferences or journals, resulting in totally 50 venues. See Table 7 for details. We removed stop words and punctuations, and maintained the most frequent 10,000 words as vocabulary.
- **Web** is a Web page hyperlink network. Each page contains the most frequent phrases of news articles and is associated with an author. Doc-doc edges are hyperlinks between pages. After removing short documents, we obtained a corpus of 445,657 documents and 36,405 authors. Again, we removed stop words and punctuations and maintained the most frequent 10,000 words as vocabulary.

A.3 Experiment Environment

All the experiments were conducted on Linux server with a Tesla K80 GPU with 11441MiB. Its operating system is CentOS Linux 7 (Core). We implemented our proposed model VGATM using Python 3.6 as programming language and TensorFlow 1.10.0 as deep learning library. Other frameworks include NumPy 1.17.4, sklearn 0.23.2, and scipy 1.5.2. We will release code and datasets upon publication.

[2] https://aylien.com/coronavirus-news-dataset/

[3] http://www.arnetminer.org/citation

[4] https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng