Layer-Assisted Neural Topic Modeling over Document Networks

Yiming Wang[1,2] , **Ximing Li**[1,2*†] , **Jihong Ouyang**[1,2*]

[1]College of Computer Science and Technology, Jilin University, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, China

{yimingw17, liximing86}@gmail.com, ouyj@jlu.edu.cn

Abstract

Neural topic modeling provides a flexible, efficient, and powerful way to extract topic representations from text documents. Unfortunately, most existing models cannot handle the text data with network links, such as web pages with hyperlinks and scientific papers with citations. To resolve this kind of data, we develop a novel neural topic model, namely Layer-Assisted Neural Topic Model (LANTM), which can be interpreted from the perspective of variational auto-encoders. Our major motivation is to enhance the topic representation encoding by not only using text contents, but also the assisted network links. Specifically, LANTM encodes the texts and network links into the topic representations by an augmented network with graph convolutional modules, and decodes them by maximizing the likelihood of the generative process. The neural variational inference is adopted for efficient inference. Experimental results validate that LANTM significantly outperforms the existing models on topic quality, text classification and link prediction.

1 Introduction

Neural topic modeling [Miao *et al.*, 2017; Srivastava and Sutton, 2017] refers to extract latent topics from text data by using Neural Variational Inference (NVI) [Miao *et al.*, 2016], which combines stochastic variational inference with deep neural networks [Kingma and Welling, 2014; Mnih and Gregor, 2014; Rezende *et al.*, 2014]. Thanks to the black-box nature of NVI, the neural topic models are flexible, efficient, and powerful for various types of model structures, beyond traditional topic models such as Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003]. As the art marrying topic modeling with deep neural networks, it has recently attracted much attention from the machine learning community [Zhang *et al.*, 2018; Dieng *et al.*, 2020; Burkhardt and Kramer, 2019; Liu *et al.*, 2019; Isonuma *et al.*, 2020; Zhu *et al.*, 2020].

Nowadays, the text data with network links are becoming pervasive in many real-world scenarios, *e.g.,* web pages with hyperlinks, scientific papers with citations, and Tweets with friendships, to name just a few. This kind of data challenges the neural topic modeling for simultaneously expressing the textual contents and network links of texts. More formally, we describe the text data with network links and the corresponding problem of neural topic modeling as follows: Consider a text document collection $\mathbf{\Omega} = \{\mathbf{X}, \mathbf{\Pi}\}$, where $\mathbf{X} = \{x_d\}_{d=1}^{D}$ and $\mathbf{\Pi} = \{\pi_{ij}\}_{i,j=1}^{D}$ denote the texts and the links between texts, respectively. Each text $x_d \in \mathbb{R}^V$ is represented by a vector of the vocabulary; and for each link, $\pi_{ij} = 1$ indicates that the texts x_i and x_j are connected, and $\pi_{ij} = 0$ otherwise. Generally, the objective of neural topic modeling over $\mathbf{\Omega}$ is to extract latent topic representations of texts, which can effectively represent content themes and maintain the network links simultaneously.

To our knowledge, there are only very few previous investigations on this subject, *e.g.,* the conventional Relational Topic Model (RTM) [Chang and Blei, 2009] and its neural variant Neural Relational Topic Model (NRTM) [Bai *et al.*, 2018]. The NRTM can be interpreted as a deep auto-encoder, where it encodes the texts into their corresponding topic representations and adopts them to separately reconstruct the texts and network links. In this work, we aim to enhance the topic representation encoding beyond NRTM by not only using text contents, but also the assisted network links. Motivated by this, we propose a novel neural topic model, namely **L**ayer-**A**ssisted **N**eural **T**opic **M**odel (**LANTM**). which can also be interpreted from the perspective of Variational Auto-Encoders (VAE). *Encoding*: We treat the network links as a text graph, therefore we design an augmented encoder network with two channels, where one is the Multi-Layer Perception (MLP) for texts and the other is the Graph Convolutional Network (GCN) for network links. To extract high-quality topic representations, the two channels work in a layer-assisted manner, where the MLP representation of each layer is aggregated with the corresponding GCN representation learned from network links. *Decoding*: We adopt the topic representations to reconstruct the texts and network links by maximizing the likelihood of the generative process of LANTM. The overall framework of LANTM is illustrated in Fig.1. We use NVI to efficiently solve LANTM. We evaluate LANTM on topic quality, text classification and

*Corresponding Author
†Contributing equally with the first author.

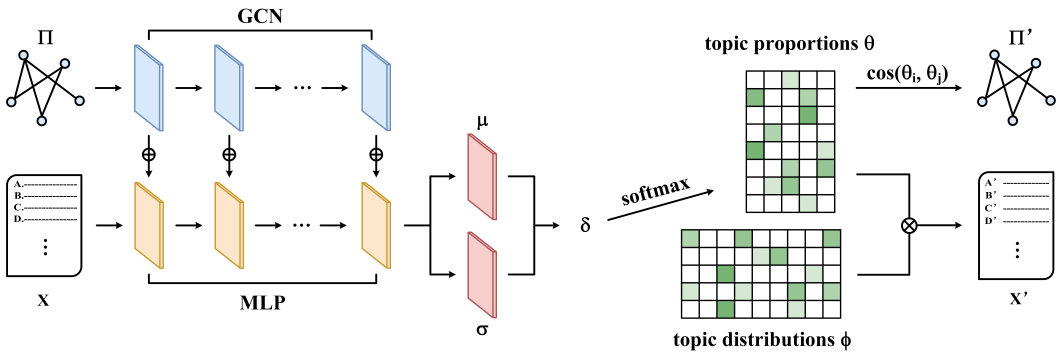

Figure 1: Overview of the model structure of LANTM.

link prediction. Experimental results validate that LANTM significantly outperforms the state-of-the-art baseline models.

To sum up, our contributions are outlined below.

1. We propose a novel neural topic model LANTM for the text data with network links, and describe its generative process and model training with NVI in **Sections 3.1 and 3.2**.
2. We develop an augmented encoder network, which jointly encodes the text content and network links into topic representations, described in **Section 3.3**.
3. We evaluate LANTM on 5 benchmark datasets. Empirical results indicate that LANTM significantly outperforms existing models, shown in **Section 4** .

2 Related Work

To our knowledge, the hierarchy of topic modeling falls into *standard probabilistic models* and *neural topic models*.

The underlying idea of standard probabilistic models supposes that the word tokens of texts are drawn from the latent distributions associated with topics. For example, the representative LDA model [Blei *et al.*, 2003] assumes that each word token is drawn from a selected topic distribution over words, and the topic is previously drawn from the topic proportions of texts. During the past decades, many extensions of LDA have been widely developed as surveyed in [Blei, 2012; Boyd-Graber *et al.*, 2017]. However, to be efficiently inferred by conventional methods [Jordan *et al.*, 1999; Griffiths and Steyvers, 2004; Mimno *et al.*, 2012; Li *et al.*, 2016], most of these methods are defined as shallow models with only three or four layers, .

Beyond LDA-based models, neural topic models [Miao *et al.*, 2017; Srivastava and Sutton, 2017; Zhang *et al.*, 2018; Dieng *et al.*, 2020; Burkhardt and Kramer, 2019; Liu *et al.*, 2019; Isonuma *et al.*, 2020; Zhu *et al.*, 2020] are solved by the generic black-box NVI [Kingma and Welling, 2014; Mnih and Gregor, 2014; Rezende *et al.*, 2014]. Therefore, they are more flexible, and allowed for more complex and deeper generative processes, growing the expressive capacity. To be specific, under the spirit of NVI, the variational distribution of latent topical variables is defined as a variational neural network that ingests texts and outputs latent topical variables. With the reparameterization technique [Kingma and Welling, 2014], the variational objective can be approximated by drawing Monte Carlo samples, and then optimized by gradient-based methods, regardless of the network architecture. Or they can be interpreted as VAE, where the variational distribution serves as an encoder network from texts to latent topical variables, and texts are reconstructed by drawing from latent distributions associated with topics as the defined generative process of the model.

Existing neural topic models mainly focus on the pure text data, however there are rare works motivated by handling the text data with network links [Bai *et al.*, 2018; Zhang and Lauw, 2020]. The NRTM [Bai *et al.*, 2018] extends the traditional RTM [Chang and Blei, 2009] by leveraging the stacked VAE. The model defines an encoder-decoder process for texts, and jointly trains a link prediction network by treating the concatenation of topic proportions as the input. Besides, the Adjacent-Encoder-X (AdjEnc-X) [Zhang and Lauw, 2020] directly treats the network links as the supplemental features of texts, and constructs a noisy autoencoder to learn topic proportions. Orthogonal to those models, LANTM treats the text content and links as bag-of-words and graph data, respectively, and encodes them by using an augmented network with graph convolutional modules. Therefore, LANTM enables to learn better topic representations that benefits from the joint learning of different types of data and the layer-assisted manner.

3 The LANTM Model

In this section, we introduce **L**ayer-**A**ssisted **N**eural **T**opic **M**odel (**LANTM**) for modeling text data with network links. We describe the generative process of LANTM as well as model training with NVI, and then interpret LANTM from the perspective of VAE.

3.1 Model Definition

Basically, LANTM can be regarded as an extension of RTM [Chang and Blei, 2009], thus its model definition is mainly inherited from RTM. We now introduce the generative process of LANTM for the content of texts and the links simultaneously. More formally, given a corpus we suppose that there exist totally K topics $\{\phi_k\}_{k=1}^{K}$, where each topic $\phi_k \in \mathbb{R}^V$ is represented by a multinomial distribution over the vocabulary. And each text is represented by a mixture of topics. To

generate a text d, it draws a topic proportion θ_d from a logistic normal distribution $\mathcal{LN}(\mu_0, \Sigma_0)$ formulated below:

$$\delta_d \sim \mathcal{N}(\mu_0, \Sigma_0), \quad \theta_d = \mathbf{softmax}(\delta_d), \tag{1}$$

where δ_d denotes the corresponding unnormalized topic proportion.[1] Each word token in text d is generated by first drawing a topic indicator z_{dn} from θ_d and then drawing x_{dn} from the selected topic $\phi_{z_{dn}}$. On the other hand, for each text pair $\{i, j\}$, it generates a link indicator π_{ij} drawn from a Bernoulli distribution parameterized by the cosine similarity between their corresponding topic proportions. The motivation coincides with the fact that the texts with similar topic proportions are more likely to share a link.

For clarity, we summarize the generative process of LANTM as follows:

1. For each text $d \in [D]$
   a. **Draw** an unnormalized topic proportion $\delta_d \sim \mathcal{N}(\mu_0, \Sigma_0)$
   b. **Compute** the topic proportion $\theta_d = \mathbf{softmax}(\delta_d)$
   c. For each word token x_{dn}, $n \in [N_d]$
      i. **Draw** a topic assignment $z_{dn} \sim \mathbf{Cat}(\theta_d)$
      ii. **Draw** a word $x_{dn} \sim \mathbf{Cat}(\phi_{z_{dn}})$
2. For each text pair $i, j \in [D]$
   a. **Draw** a link indicator $\pi_{ij} \sim \mathbf{Bernoulli}(\mathbf{cos}(\theta_i, \theta_j))$

By revisiting the model definitions of RTM [Chang and Blei, 2009] and LANTM, we show that the major difference is the generative ways of the topic proportion θ per-text, which can be interpreted as the latent representation of text. As a neural topic model with NVI, in LANTM the topic proportion θ can be inferred by more flexible and powerful encoder network based on the observations of word tokens and links, beyond RTM that directly generates θ from a Dirichlet distribution. We will detail the model training process and network architecture in the following subsections.

3.2 Model Training

Given an observation of collection $\mathbf{\Omega} = \{\mathbf{X}, \mathbf{\Pi}\}$ with texts $\mathbf{X} = \{x_d\}_{d=1}^D$ and links $\mathbf{\Pi} = \{\pi_{ij}\}_{i,j=1}^D$, the objective of LANTM training is to estimate the latent variables of interest, including the unnormalized topic proportions of texts δ and topic distributions ϕ. We neglect the topic assignments z since it can be analytically integrated out. Commonly, the model training can be achieved by maximizing the following log marginal likelihood of $\mathbf{X}$ and $\mathbf{\Pi}$:

$$\begin{aligned}\mathcal{L}(\delta, \phi) =& \sum_{d=1}^{D} \log p(\theta_d) p(x_d|\theta_d, \phi) + \sum_{i,j=1}^{D} \log p(\pi_{ij}|\theta_i, \theta_j) \\ =& \sum_{d=1}^{D} \log p(\delta_d) p(x_d|\delta_d, \phi) + \sum_{i,j=1}^{D} \log p(\pi_{ij}|\delta_i, \delta_j)\end{aligned} \tag{2}$$

The above formula is intractable to maximize, since it involves a difficult integral over the (unnormalized) topic proportions. Accordingly, we resort to approximate training by leveraging the amortized variational inference [Gershman and Goodman, 2014]. To be specific, we define the variational distribution that depends on both observations $\{x_d, \pi_d\}$ and the variational parameter γ, formulated below:

$$q(\delta; \mathbf{X}, \mathbf{\Pi}, \gamma) = \sum_{d=1}^{D} q(\delta_d; x_d, \pi_d, \gamma) \tag{3}$$

For each δ_d, the corresponding variational distribution is a Gaussian whose mean μ_d and covariance Σ_d are the outputs of the encoder network parameterized by γ. In other words, the variational distribution can be interpreted as an encoder network that ingests $\{x_d, \pi_d\}$ and outputs the mean and covariance of δ_d.

We apply this family of variational distribution, thus under the spirit of NVI we formulate the following lower bound of Eq.(2), *i.e.,* a variational objective of LANTM with respect to the topic distributions ϕ and variational parameter γ:

$$\begin{aligned}\mathcal{L}(\phi, \gamma) =& \sum_{d=1}^{D} \mathbb{E}_q \left[\log p(x_d|\delta_d, \phi)\right] + \sum_{i,j=1}^{D} \mathbb{E}_q \left[\log p(\pi_{ij}|\delta_i, \delta_j)\right] \\ &- \sum_{d=1}^{D} \mathrm{KL}\left[q(\delta_d|x_d, \pi_d, \gamma) \,||\, p(\delta_d)\right].\end{aligned} \tag{4}$$

The conditional distributions within the first two terms can be presented as follows:

$$p(x_d|\delta_d, \phi) = \prod_{n=1}^{N_d} \sum_{k=1}^{K} \theta_{dk} \phi_{k x_{dn}} \tag{5}$$

$$p(\pi_{ij}|\delta_i, \delta_j) = \lambda^{\pi_{ij}} (1-\lambda)^{(1-\pi_{ij})} \tag{6}$$

where $\lambda = \mathbf{cos}(\theta_i, \theta_j) = \mathbf{cos}(\delta_i, \delta_j)$.

Unfortunately, the variational objective of Eq.(4) is still intractable to maximize since it involves the expectations with respect to the (unknown) target variational distribution. To this end, we form a Monte Carlo approximation to the variational objective as follows:

$$\begin{aligned}\mathcal{L}(\phi, \gamma) \approx& \frac{1}{S} \sum_{d=1}^{D} \sum_{s=1}^{S} \log p(x_d|\delta_d^{(s)}, \phi) \\ &+ \frac{1}{S} \sum_{i,j=1}^{D} \sum_{s=1}^{S} \log p(\pi_{ij}|\delta_i^{(s)}, \delta_j^{(s)}) \\ &- \sum_{d=1}^{D} \mathrm{KL}\left[q(\delta_d|x_d, \pi_d, \gamma) \,||\, p(\delta_d)\right]. \\ & \qquad \delta_d^{(s)} \sim q(\delta_d; x_d, \pi_d, \gamma) \quad s \in [S]\end{aligned} \tag{7}$$

where S is the number of samples; and the samples are exactly generated by leveraging the reparameterization trick [Kingma and Welling, 2014]:

$$\delta_d^{(s)} = \mu_d + \Sigma_d^{\frac{1}{2}} \epsilon_d^{(s)}, \qquad \epsilon_d^{(s)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{8}$$

Besides, the KL-divergence regularization per-text has an analytic form, which can be expanded as follows:

$$\begin{aligned}&\mathrm{KL}\left[q(\delta_d|x_d, \pi_d, \gamma) \,||\, p(\delta_d)\right] = \\ &\qquad \frac{1}{2}(\mathbf{Tr}(\Sigma_d) + \mu_d^\top \mu_d - \log \mathbf{det}(\Sigma_d) - K),\end{aligned} \tag{9}$$

[1] Following [Dieng *et al.*, 2020] we fix the Gaussian prior $\mathcal{N}(\mu_0, \Sigma_0)$ as the standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

Algorithm 1 Training process for LANTM

Input: $\mathbf{X} \in \mathbb{R}^{D\times V}$: bag-of-words matrix; $\mathbf{\Pi} \in \mathbb{R}^{D\times D}$: text links matrix; K: number of topics.
Output: topic proportions of texts θ, topic distributions ϕ

```
Initialize network parameters W^(g), W, b and topic distributions φ normally;
for epoch = 1 to maxEpoch do
    for n = 1 to N − 1 do
        Calculate GCN layer H_n = ψ(Π̃ H_{n−1} W_n^(g)) ;
        Calculate MLP layer Z_n = ψ(Ẑ_{n−1} W_n + b_n);
        Combine two layers: Ẑ_n = ξ H_n + (1 − ξ) Z_n;
    end for
    Calculate mean and variance by Eq.(13);
    Calculate topic proportions θ by Eq.(14);
    Calculate the gradients of {φ, γ} by backpropagation;
    Update {φ, γ} with Adam;
end for
Return θ and φ.
```

where $\mathbf{Tr}(\cdot)$ and $\mathbf{det}(\cdot)$ denote the trace and determinant of a matrix, respectively. Accordingly, our LANTM can be approximately inferred by forming the gradients of Eq.(7) with respect to $\{\phi, \gamma\}$, and updating them with any adaptive learning rate method.

3.3 Interpreting LANTM as VAE

We can interpret LANTM from the perspective of VAE, since the NVI method is adopted for model training. Revisiting Eq.(4), the first two terms play the roles of the reconstruction errors of observations. In other words, the variational distribution serves as the encoder that encodes the texts and network links to (unnormalized) topic proportions of texts $\{\delta, \theta\}$, and LANTM reconstructs the observations by maximizing their log marginal likelihood of the generative formulation given $\{\delta, \theta\}$. The overall framework of LANTM is illustrated in Fig.1. We now describe the encoder architecture in more detail.

Encoder architecture. We declare that $\mathbf{X} \in \mathbb{R}^{D\times V}$ and $\mathbf{\Pi} \in \mathbb{R}^{D\times D}$ denote the bag-of-words matrix and text link matrix, respectively. Let N denotes the number of layers in the encoder network. For the former $N-1$ layers, each of the n-th layer representations of the channels of GCN and MLP are described as follows:

$$\mathbf{H}_n = \psi(\tilde{\mathbf{\Pi}}\mathbf{H}_{n-1}\mathbf{W}_n^{(g)}), \quad \mathbf{Z}_n = \psi(\mathbf{Z}_{n-1}\mathbf{W}_n + \mathbf{b}_n), \tag{10}$$

where $\tilde{\mathbf{\Pi}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{\Pi}\mathbf{D}^{-\frac{1}{2}}$ is the symmetrically normalization of $\mathbf{\Pi}$; $\psi(\cdot)$ represents the Sigmoid activation function; $\mathbf{W}_n^{(g)}$, $\mathbf{W}_n$ and $\mathbf{b}_n$ are network weights and bias parameters of the n-th layer. The first layers are defined as:

$$\mathbf{H}_0 = \psi(\tilde{\mathbf{\Pi}}\mathbf{X}\mathbf{W}_0^{(g)}), \quad \mathbf{Z}_0 = \psi(\mathbf{X}\mathbf{W}_0 + \mathbf{b}_0). \tag{11}$$

Inspired by [Bo *et al.*, 2020], we employ the layer-wise assistance, which combines each $\mathbf{Z}_n$ and $\mathbf{H}_n$ with a combining coefficient ξ and sets the combined latent representation $\hat{\mathbf{Z}}_n$ as the input of the next MLP layer:

$$\hat{\mathbf{Z}}_n = \xi\mathbf{H}_n + (1-\xi)\mathbf{Z}_n, \quad \mathbf{Z}_{n+1} = \psi(\hat{\mathbf{Z}}_n\mathbf{W}_{n+1} + \mathbf{b}_{n+1}). \tag{12}$$

Finally, the last layer outputs the mean and covariance of δ as follows:

$$\mu = \psi(\mathbf{Z}_{N-1}\mathbf{W}_\mu + \mathbf{b}_\mu), \quad \mathbf{\Sigma} = \psi(\mathbf{Z}_{N-1}\mathbf{W}_{\mathbf{\Sigma}} + \mathbf{b}_{\mathbf{\Sigma}}). \tag{13}$$

Referring to Eq.(8), the topic proportions of texts θ are computed by using the Monte Carlo samples:

$$\theta = \mathbf{softmax}(\mu + \mathbf{\Sigma}\epsilon), \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{14}$$

3.4 Training and Inference Summary

Model training. Following [Kipf and Welling, 2016], we set three layers for both channels of MLP and GCN. The sub-gradients of the encoder networks are computed by backpropagation, the Adam [Kingma and Ba, 2015] is applied for adaptively setting the learning rate. The overall training details of LANTM are summarized in *Algorithm 1*.

Inference for future texts. We directly use the fitted encoder network to compute the topic proportions of future texts. In practice, the network links of future texts may be unknown. To resolve this, following [Zhang and Lauw, 2020] we reuse the training texts as auxiliary data. Specifically, we apply the known text link matrix between training texts and future ones or construct the *k*-nearest neighbor matrix between them instead. We feed all texts and the auxiliary link matrix into the fitted encoder network to compute the topic proportions, and leave the ones of future texts only.

4 Experiment

Datasets. In the experiments, we apply the dataset of Cora[2] consisting of paper abstracts and citations [McCallum *et al.*, 2000], and *Reuters*[3] (*R8*) without any links. The Cora dataset is divided into four subsets, namely *Data Structure* (*DS*), *Hardware and Architecture* (*HA*), *Machine Learning* (*ML*) and *Programming Language* (*PL*). We apply Cora processed by [Zhang and Lauw, 2020]. For *R8*, the standard stop words and infrequent words occurring in less than 5 documents are filtered out, and a *k*-nearset neighbor graph is constructed ($k = 10$). Statistics of datasets are presented in Table 1.

Comparing models. We totally select 7 existing baseline models for comparison, including 3 standard neural topic models, **NVDM**[4] [Miao *et al.*, 2016], **ProdLDA**[5] [Srivastava and Sutton, 2017], **ETM**[6] [Dieng *et al.*, 2020] and 4 methods handling data with network links, **RTM**[7] [Chang and Blei, 2009], **NRTM**[8] [Bai *et al.*, 2018], **VGAE**[9] [Kipf and Welling, 2016], **AdjEnc-X**[10] [Zhang and Lauw, 2020].

For our LANTM, the combining coefficient ξ is tuned over $\{0.1, 0.2, \ldots, 0.9\}$. For all baseline models, the default parameters are adopted. All methods are trained under same num of epochs and the topic numbers are set as $\{25, 50\}$ for all datasets.

[2] http://people.cs.umass.edu/mccallum/data/cora-classify.tar.gz
[3] https://martin-thoma.com/nlp-reuters/
[4] https://github.com/ysmiao/nvdm
[5] https://github.com/akashgit/autoencoding_vi_for_topic_models
[6] https://github.com/adjidieng/ETM
[7] http://cran.r-project.org/web/packages/lda/
[8] https://github.com/zbchern/Neural-Relational-Topic-Models
[9] https://github.com/tkipf/gae
[10] https://github.com/PreferredAI/adjacent-encoder

Dataset	#doc	#train	#test	#word	#link	AvgL	#class
DS	570	456	114	3009	1336	67.8	9
HA	223	178	45	2023	515	77.1	6
ML	1980	1584	396	4265	5748	68.1	7
PL	1552	1241	311	3945	4851	70.3	9
R8	7558	6046	1512	4166	-	53.8	8

Table 1: Summary of dataset statistics. "AvgL" denotes the average document length.

Method	K	DS	HA	ML	PL	R8
LANTM	25	**0.48**	**0.44**	**0.51**	**0.49**	0.45
	50	**0.49**	**0.44**	**0.52**	**0.50**	0.45
NVDM	25	0.43	0.41	0.44	0.42	0.45
	50	0.43	0.41	0.45	0.43	0.44
ProdLDA	25	0.42	0.43	0.44	0.43	0.43
	50	0.43	**0.44**	0.44	0.43	0.43
ETM	25	0.39	0.39	0.39	0.39	0.36
	50	0.39	0.39	0.39	0.39	0.35
RTM	25	0.38	0.39	0.38	0.40	0.45
	50	0.38	0.37	0.38	0.39	0.45
NRTM	25	0.45	**0.44**	0.47	0.45	0.45
	50	0.45	0.42	0.48	0.45	0.45
AdjEnc-X	25	0.47	**0.44**	0.49	0.48	**0.48**
	50	0.47	**0.44**	0.51	0.49	**0.47**

Table 2: Experimental results of TC. The higher score means better performance, and the best scores are in boldface.

4.1 Evaluation of Topic Coherence

Topic Coherence (TC) is a popular metric to measure the topic quality by calculating co-occurrences of top-k topical words over an external corpus. In the experiment, we employ the public TC project *Palmetto*[11] [Röder *et al.*, 2015], and the setting of C_V is applied. We present the TC scores of top-10 topical words in Table 2.

We can observe that our LANTM outperforms baseline models in most settings, directly indicating LANTM can generate more effectively coherent topical words. For example, our LANTM is about 0.02 higher than baselines on DS when $K = 50$. The previous literature [Bai *et al.*, 2018] reports that a shallower decoder limits topic quality performance, and we exactly observe that the baseline models ProdLDA and NVDM with shallower decoders perform worse than NRTM with deeper decoder. We kindly notice that LANTM is also with one-layer decoder, but it significantly performs better than NRTM. Therefore the results provide strong evidence to the effectiveness of layer-assisted encoder for generating coherent topics.

4.2 Evaluation of Classification

We compare LANTM against baseline models by classification. For each comparing model, the learned topic proportions of texts are used to train the SVMs classifier.[12] In both transductive and inductive settings, we conduct 5-fold cross-validation experiments, and report the average scores of Micro-F1 and Macro-F1 in Table 3.

[11]https://github.com/dice-group/Palmetto/wiki/Coherences

[12]http://scikit-learn.org/

Overall, we find that LANTM outperforms the baseline models, and achieves significant improvement in many cases. For example, the performance gain is about 0.22 and 0.28 on PL and R8, respectively. The results directly indicate the great advantage of LANTM on learning discriminative latent topic representations. More importantly, it can be seen that LANTM also beats baseline models in the inductive setting, empirically suggesting that LANTM enables to effectively fit future text data.

4.3 Evaluation of Link Prediction

Link prediction measures prediction capacity for unseen links. For each link π_{ij}, we predict it by using the corresponding topic proportions $\{\theta_i, \theta_j\}$ to estimate the probability $P(\pi_{ij} = 1|\theta_i, \theta_j) \propto exp(-\|\theta_i - \theta_j\|^2)$. In the transductive setting, following [Kipf and Welling, 2016; Zhang and Lauw, 2020] we randomly remove one link for the texts with more than 3 links, and predict the removed ones. In the inductive setting, we randomly select 80% texts and the corresponding links for training, and directly predict all links of the remaining 20% texts. We evaluate the results by Area Under the ROC Curve (AUC) and Average Precision (AP), computed by referring to [Kipf and Welling, 2016]. We report the average scores of 5 independent runs in Table 4.

In this evaluation, our LANTM consistently outperforms baseline models in the transductive setting, and ranks the first in most cases of inductive setting. In terms of ML, it achieves the highest improvements, *i.e.,* about 0.160 and 0.147 on AUC and AP when $K = 25$. Besides, we observe that the link-based models RTM, NRTM, and AdjEnc-X almost perform better than the traditional models NVDM, ProdLDA, and ETM. This demonstrates the positive effect of network links for model fitting.

4.4 Parameter Evaluation

In this section, we evaluate the impacts of topic number K and combining coefficient ξ by TC and Macro-F1 scores in the transductive setting and plot results in Fig.2.

For the topic number K, we vary it from the set of $\{25, 50, 75, 100, 125\}$. We find that there exists a slight rising trend for TC on most datasets and best results mostly lies in $K = 75$. As for Macro-F1, our LANTM shows insensitivity on topic number which makes it practical in real applications.

For combining coefficient ξ, we vary it from an increasing set $\{0.1, 0.2, \ldots, 0.9\}$. We find that LANTM has the best TC when $\xi = 0.1$ on most datasets. The reason may goes to the MLP module for document context matrix plays more important roles on extracting coherent topic representations. Meanwhile Macro-F1 shows rising trends and best scores achieve at $\xi = 0.6$ and 0.8 for most cases. This indicates that the GCN module may contribute to learning more discriminative latent topic proportions.

5 Conclusion

In this paper, we propose a novel encoder-augmented topic model LANTM for combining normal bag-of-words text data with ubiquitous network links to jointly learn latent topic representations. MLP and GCN are applied on these data in diverse structures and we employ layer-wise augmentation for

Model	K	Transductive Learning										Inductive Learning									
		Micro-F1					Macro-F1					Micro-F1					Macro-F1				
		DS	HA	ML	PL	R8	DS	HA	ML	PL	R8	DS	HA	ML	PL	R8	DS	HA	ML	PL	R8
LANTM	25	**.739**	**.846**	**.847**	**.751**	**.965**	**.674**	.806	**.825**	**.688**	**.909**	**.544**	**.711**	**.765**	**.679**	**.942**	**.407**	**.579**	**.673**	**.552**	**.841**
	50	**.746**	**.857**	**.851**	**.756**	**.961**	**.694**	**.813**	**.828**	**.687**	**.893**	**.527**	**.711**	**.747**	**.665**	**.940**	**.358**	**.605**	**.678**	**.558**	**.829**
NVDM	25	.396	.701	.403	.374	.775	.314	.589	.342	.263	.543	.254	.444	.230	.299	.572	.150	.218	.165	.107	.212
	50	.480	.740	.545	.460	.827	.389	.649	.471	.348	.647	.236	.422	.280	.286	.661	.144	.156	.194	.108	.298
ProdLDA	25	.457	.709	.596	.464	.805	.330	.502	.511	.354	.441	.308	.467	.399	.383	.667	.242	.303	.276	.198	.217
	50	.437	.730	.601	.483	.825	.333	.570	.508	.372	.493	.369	.467	.460	.424	.539	.246	.210	.331	.232	.111
ETM	25	.514	.761	.574	.486	.851	.420	.599	.502	.386	.584	.325	.578	.391	.421	.700	.243	.374	.308	.267	.297
	50	.553	.754	.622	.530	.871	.467	.676	.566	.444	.683	.306	.511	.490	.389	.527	.231	.342	.367	.245	.185
RTM	25	.585	.781	.643	.524	.869	.490	.630	.583	.439	.616	.140	.467	.167	.225	.506	.118	.184	.128	.094	.085
	50	.588	.799	.686	.580	.895	.512	.701	.647	.507	.689	.106	.400	.174	.270	.506	.053	.166	.159	.106	.086
NRTM	25	.632	.836	.672	.599	.839	.566	**.822**	.631	.516	.623	.438	.667	.556	.466	.858	.314	.506	.507	.308	.603
	50	.641	.807	.664	.588	.826	.571	.765	.620	.505	.583	.465	.556	.520	.476	.823	.319	.336	.434	.322	.577
VGAE	25	.576	.682	.371	.494	.861	.494	.550	.310	.420	.759	.274	.511	.117	.146	.764	.228	.207	.106	.102	.618
	50	.549	.648	.337	.471	.841	.463	.465	.266	.389	.742	.288	.471	.046	.154	.768	.234	.182	.036	.104	.622
AdjEnc-X	25	.621	.804	.722	.636	.941	.539	.680	.656	.554	.842	.360	.644	.722	.560	.915	.234	.445	.621	.439	.708
	50	.718	.837	.826	.734	.951	.631	.779	.792	.663	.879	.421	.533	.745	.592	.933	.239	.329	.664	.474	.770

Table 3: Classification results of Micro-F1 and Macro-F1. The higher score means better performance, and the best scores are in boldface.

Model	K	Transductive Learning										Inductive Learning									
		AUC					AP					AUC					AP				
		DS	HA	ML	PL	R8	DS	HA	ML	PL	R8	DS	HA	ML	PL	R8	DS	HA	ML	PL	R8
LANTM	25	**.903**	**.879**	**.885**	**.870**	.948	**.900**	**.879**	**.893**	**.890**	.942	.827	**.876**	**.821**	**.749**	.903	.827	**.837**	**.836**	**.745**	**.909**
	50	**.872**	.839	**.854**	**.848**	.924	**.879**	.852	**.866**	**.870**	.920	**.899**	.839	**.795**	**.735**	.880	**.910**	.811	**.804**	**.711**	.881
NVDM	25	.634	.694	.569	.597	.722	.622	.671	.556	.574	.689	.374	.575	.505	.474	.589	.445	.539	.490	.456	.565
	50	.658	.739	.614	.637	.737	.650	.726	.597	.615	.708	.498	.646	.523	.517	.617	.499	.581	.515	.509	.600
ProdLDA	25	.724	.749	.699	.690	.778	.706	.721	.672	.658	.740	.491	.531	.637	.576	.686	.562	.612	.629	.565	.674
	50	.727	.739	.689	.693	.788	.715	.712	.664	.670	.755	.399	.626	.613	.583	.663	.443	.639	.610	.588	.650
ETM	25	.736	.777	.670	.688	.855	.746	.765	.676	.698	.853	.838	.673	.687	.653	.781	.825	.707	.708	.654	.776
	50	.712	.736	.653	.677	.822	.726	.738	.667	.695	.827	.701	.683	.648	.575	.729	.644	.735	.650	.614	.743
RTM	25	.811	.823	.725	.710	.623	.825	.837	.746	.736	.689	.412	.537	.569	.473	.495	.484	.652	.544	.496	.511
	50	.781	.795	.730	.695	.622	.808	.801	.761	.729	.702	.396	.675	.512	.434	.487	.448	.693	.512	.452	.507
NRTM	25	.786	.843	.689	.762	.792	.793	.824	.710	.769	.781	**.859**	.861	.733	.697	.721	**.879**	.831	.726	.686	.708
	50	.756	.774	.679	.737	.726	.779	.769	.696	.749	.717	.885	**.873**	.715	.675	.820	.905	**.860**	.701	.678	.782
VGAE	25	.688	.630	.507	.612	.722	.758	.669	.573	.676	.794	.593	.605	.611	.622	.599	.634	.613	.630	.632	.648
	50	.620	.619	.514	.601	.711	.703	.646	.569	.655	.787	.623	.617	.618	.627	.604	.651	.622	.625	.650	.649
AdjEnc-X	25	.738	.852	.624	.653	**.982**	.733	.850	.657	.689	**.980**	.695	.559	.751	.743	**.904**	.671	.640	.751	.717	.899
	50	.845	**.901**	.750	.798	**.993**	.850	**.885**	.789	.834	**.993**	.738	.772	.727	.695	**.922**	.716	.809	.730	.684	**.927**

Table 4: Link prediction results of AUC and AP. The higher score means better performance, and the best scores are in boldface.

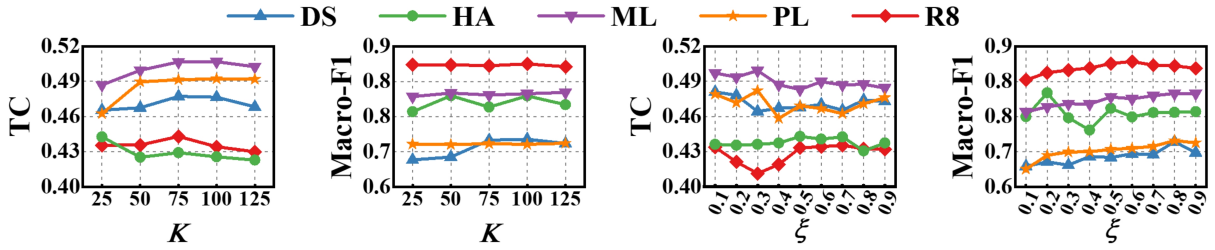

Figure 2: Parameter evaluation results on TC and Macro-F1 by varying K (left section) and ξ (right section).

combining each layer. Text data and network links are reconstructed separately according to different structures. Empirical studies on three commonly acknowledged metrics demonstrate our significant improvement against existing representative methods.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (NSFC) (No.61876071) and Scientific and Technological Developing Scheme of Jilin Province (No.20180201003SF, No.20190701031GH) and Energy Administration of Jilin Province (No.3D516L921421).

References

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

[Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

[Blei, 2012] David M Blei. Probabilistic topic models. *Communications of The ACM*, 55(4):77–84, 2012.

[Bo *et al.*, 2020] Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural deep clustering network. In *The Web Conference*, pages 1400–1410, 2020.

[Boyd-Graber *et al.*, 2017] Jordan Boyd-Graber, Yuening Hu, and David Mimno. Applications of topic models. *Foundations and Trends in Information Retrieval*, 11(2-3):143–296, 2017.

[Burkhardt and Kramer, 2019] Sophie Burkhardt and Stefan Kramer. Decoupling sparsity and smoothness in the Dirichlet variational autoencoder topic model. *Journal of Machine Learning Research*, 20(131):1–27, 2019.

[Chang and Blei, 2009] Jonathan Chang and David M. Blei. Relational topic models for document networks. In *International Conference on Artificial Intelligence and Statistics*, pages 81–88, 2009.

[Dieng *et al.*, 2020] Adji B Dieng, Francisco J R Ruiz, and David M Blei. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453, 2020.

[Gershman and Goodman, 2014] Samuel Gershman and Noah D. Goodman. Amortized inference in probabilistic reasoning. In *Annual Meeting of the Cognitive Science Society*, 2014.

[Griffiths and Steyvers, 2004] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(suppl 1):5228–5235, 2004.

[Isonuma *et al.*, 2020] Masaru Isonuma, Junichiro Mori, Danushka Bollegala, and Ichiro Sakata. Tree-Structured Neural Topic Model. In *Annual Meeting of the Association for Computational Linguistics*, pages 800–806, 2020.

[Jordan *et al.*, 1999] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):105–161, 1999.

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[Kipf and Welling, 2016] Thomas N. Kipf and Max Welling. Variational graph auto-encoders. In *NIPS Workshop on Bayesian Deep Learning*, 2016.

[Li *et al.*, 2016] Ximing Li, Jihong Ouyang, and Xiaotang Zhou. Sparse hybrid variational-Gibbs algorithm for latent Dirichlet allocation. In *SIAM International Conference on Data Mining*, pages 729–737, 2016.

[Liu *et al.*, 2019] Luyang Liu, Heyan Huang, Yang Gao, Yongfeng Zhang, and Xiaochi Wei. Neural variational correlated topic modeling. In *The Web Conference*, pages 1142–1152, 2019.

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

[Miao *et al.*, 2016] Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In *International Conference on Machine Learning*, pages 1727–1736, 2016.

[Miao *et al.*, 2017] Yishu Miao, Edward Grefenstette, and Phil Blunsom. Discovering discrete latent topics with neural variational inference. In *International Conference on Machine Learning*, pages 2410–2419, 2017.

[Mimno *et al.*, 2012] David M. Mimno, Matthew D. Hoffman, and David M. Blei. Sparse stochastic inference for latent Dirichlet allocation. In *International Conference on Machine Learning*, page 1515–1522, 2012.

[Mnih and Gregor, 2014] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799, 2014.

[Rezende *et al.*, 2014] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

[Röder *et al.*, 2015] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *International Conference on Web Search and Data Mining*, page 399–408, 2015.

[Srivastava and Sutton, 2017] Akash Srivastava and Charles A. Sutton. Autoencoding variational inference for topic models. In *International Conference on Learning Representations*, 2017.

[Zhang and Lauw, 2020] Ce Zhang and Hady W. Lauw. Topic modeling on document networks with adjacent-encoder. In *AAAI Conference on Artificial Intelligence*, 2020.

[Zhang *et al.*, 2018] Hao Zhang, Bo Chen, Dandan Guo, and Mingyuan Zhou. WHAI: Weibull hybrid autoencoding inference for deep topic modeling. In *International Conference on Learning Representations*, 2018.

[Zhu *et al.*, 2020] Qile Zhu, Zheng Feng, and Xiaolin Li. Graph attention topic modeling network. In *The Web Conference*, pages 1142–1152, 2020.