5-2023

Document graph representation learning

Ce ZHANG
Singapore Management University, cezhang@smu.edu.sg

DOCUMENT GRAPH REPRESENTATION LEARNING

CE ZHANG

SINGAPORE MANAGEMENT UNIVERSITY
2023

Document Graph Representation Learning

by
Ce Zhang

Submitted to School of Computing and Information Systems in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Computer Science

Dissertation Committee:

Hady W. Lauw (Supervisor/Chair)
Associate Professor
Singapore Management University

Baihua Zheng
Professor
Singapore Management University

Jing Jiang
Professor
Singapore Management University

Jiliang Tang
University Foundation Professor
Michigan State University

Singapore Management University
2023

I hereby declare that this PhD dissertation is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in this dissertation.

This PhD dissertation has also not been submitted for any degree in any university previously.

Ce Zhang

20 April 2023

Document Graph Representation Learning

Ce Zhang

Abstract

Much of the data on the Web can be represented in a graph structure, ranging from social and biological to academic and Web page graphs, etc. Graph analysis recently attracts escalating research attention due to its importance and wide applicability. Diverse problems could be formulated as graph tasks, such as text classification and information retrieval. As the primary information is the inherent structure of the graph itself, one promising direction known as the *graph representation learning* problem is to learn the representation of each node, which could in turn fuel tasks such as node classification, node clustering, and link prediction.

As a specific graph data, documents are usually connected in a graph structure. For example, Google Web pages hyperlink to other related pages, academic papers cite other papers, Facebook user profiles are connected as a social network, news articles with similar tags are linked together, etc. We call such data *document graph* or *document network*. To better make sense of the meaning within these text documents, researchers develop neural topic models. By modeling both textual content within documents and connectivity across documents, we can discover more interpretable topics to understand the corpus and better fulfill real-world applications, such as Web page searching, news article classification, academic paper indexing, and friend recommendation based on user profiles, etc. However, traditional topic models explore the content only, ignoring the connectivity. In this dissertation, we aim to develop models for document graph representation learning.

First, we investigate the extension of Auto-Encoders, a family of shallow topic models. Intuitively, connected documents tend to share similar latent topics. Thus, we allow Auto-Encoder to extract topics of the input document and reconstruct its adjacent neighbors. This allows doc-

uments in a network to collaboratively learn from one another, such that close neighbors would have similar representations in the topic space. Extensive experiments verify the effectiveness of our proposed model against both graphical and neural baselines.

Second, we focus on dynamic modeling of document networks. In many real-world scenarios, documents are published in a sequence and are associated with timestamps. For example, academic papers published over the years exhibit the development of research topics. To incorporate such temporal information, we introduce a neural topic model aimed at learning unified topic distributions that incorporate both document dynamics and network structure.

Third, we discover that 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. We propose a Variational Graph Author Topic Model for documents to integrate both topic modeling and authorship and venue modeling into a unified framework.

Fourth, most previous topic models treat documents of different lengths uniformly, assuming that each document is sufficiently informative. However, shorter documents may have only a few word co-occurrences, resulting in inferior topic quality. Some other previous works assume that all documents are short, and leverage external auxiliary data, e.g., pretrained word embeddings and document connectivity. Orthogonal to existing works, we remedy this problem within the corpus itself by meta-learning and proposing a Meta-Complement Topic Model, which improves topic quality of short texts by transferring the semantic knowledge learned on long documents to complement semantically limited short texts.

Fifth, we explore the modeling of short texts on the graph. Text embedding models usually rely on word co-occurrences within the documents to learn effective representations. However, short texts with only a few words may influence the learning process. To accurately discover the main topics of these short documents, we propose a new statistical concept, i.e., optimal transport

barycenter, to incorporate external knowledge, such as pre-trained word embedding on a large corpus, to improve topic modeling. The proposed model shows state-of-the-art performance.

Contents

List of Publications

Publications

1. **Delvin Ce Zhang**, Rex Ying, and Hady W. Lauw, "Hyperbolic Graph Topic Modeling Network with Continuously Updated Topic Tree". In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (**KDD-23**), 2023.
2. **Delvin Ce Zhang** and Hady W. Lauw, "Meta-Complementing the Semantics of Short Texts in Neural Topic Models". In Proceedings of the 36th Conference on Neural Information Processing Systems (**NeurIPS-22**), 2022.
3. **Delvin Ce Zhang** and Hady W. Lauw, "Variational Graph Author Topic Modeling". In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (**KDD-22**), 2022.
4. **Delvin Ce Zhang** and Hady W. Lauw, "Dynamic Topic Models for Temporal Document Networks". In Proceedings of the 39th International Conference on Machine Learning (**ICML-22**), 2022.
5. **Delvin Ce Zhang** and Hady W. Lauw, "Topic Modeling for Multi-Aspect Listwise Comparisons". In Proceedings of the 30th ACM International Conference on Information and Knowledge Management (**CIKM-21**), 2021.
6. **Delvin Ce Zhang** and Hady W. Lauw, "Semi-Supervised Semantic Visualization on Networked Documents". In Proceedings of the European Conference on Machine Learning

and Principles and Practice of Knowledge Discovery in Databases (**ECML/PKDD-21**), 2021.

7. **Delvin Ce Zhang** and Hady W. Lauw, "Representation Learning on Multi-Layered Heterogeneous Network". In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (**ECML/PKDD-21**), 2021.
8. **Ce Zhang** and Hady W. Lauw. "Topic Modeling on Document Networks with Adjacent-Encoder". In Proceedings of the 34th AAAI Conference on Artificial Intelligence (**AAAI-20**), 2020.

Working papers

1. **Delvin Ce Zhang** and Hady W. Lauw, "Topic Modeling on Document Networks with Dirichlet Optimal Transport Barycenter". **Being reviewed**.
2. **Delvin Ce Zhang**, Hady W. Lauw, and Rex Ying, "Interpretable Molecule Representation Learning from Molecular Structures and Texts". **Being reviewed**.

List of Figures

List of Tables

List of Notations

Notation	Description
$\mathcal{G}$	a document network used in Chapter 3, 4, 6, and 7.
$\mathcal{D}$	a corpus of documents used in all chapters, $\mathcal{D} = \{\mathbf{d}_i\}_{i=1}^{N}$
N	number of documents in the corpus used in all chapters, $N = \|\mathcal{D}\|$.
$\mathcal{E}$	a set of edges used in Chapter 3, 6, and 7.
$\mathcal{E}$	a set of adjacency matrices used in Chapter 4, $\mathcal{E} = \{\mathcal{E}_t\}_{t=1}^{T}$, $\mathcal{E}_t \in \mathbb{R}^{N \times N}$.
T	maximum timestamp observed in training set used in Chapter 4.
$\mathbf{d}$	a document representation in the word space, $\mathbf{d} \in \mathbb{R}^{\|\mathcal{V}\|}$ used in all chapters.
$\mathcal{V}$	vocabulary used in Chapter 3, 4, 6, and 7.
$\mathcal{N}(i)$	the neighbor set of document i used in Chapter 3, 6, and 7.
$\mathcal{N}_t(i)$	the neighbor set of document i at time t used in Chapter 4.
$\mathcal{T}$	a set of timestamps used in Chapter 4, $\mathcal{T} = \{t_i\}_{t=1}^{N}$.
$\mathbf{z}_i$	topic distribution of document i used in all chapters, $\mathbf{z}_i \in \mathbb{R}^K$.
K	number of topics used in all chapters.
a_{ij}	attention value between document i and j used in all chapters.
$\mathbf{h}_t$	time embedding of timestamp t used in Chapter 4, $\mathbf{h}_t \in \mathbb{R}^{2K}$.
$\mathbf{C}$	cost matrix of time-aware OT used in Chapter 4, $\mathbf{C} \in \mathbb{R}^{T \times K \times \|\mathcal{V}\|}$
$\mathbf{P}$	transport plan of time-aware OT used in Chapter 4
$\mathbf{g}_{tk}$	topic embedding of topic k at timestamp t used in Chapter 4

$\mathbf{e}_w$	word embedding of word w used in Chapter 4 and 7
$\mathcal{A}$	a set of authors used in Chapter 5.
$\mathcal{V}$	a set of publication venues used in Chapter 5.
$\mathcal{X}$	edge connections among documents used in Chapter 5.
$\mathcal{G}$	a hierarchical multi-layered document graph based on corpus $\mathcal{C}$ used in Chapter 5.
$\mathcal{U}$	a set of vertices of $\mathcal{G}$ used in Chapter 5, we have $\mathcal{U} = \mathcal{D} \cup \mathcal{W} \cup \mathcal{A} \cup \mathcal{V}$.
$\mathcal{E}$	a set of edges of $\mathcal{G}$ used in Chapter 5, we have $\mathcal{X} \subseteq \mathcal{E}$.
$\mathcal{O}$	a set of vertex types used in Chapter 5.
$\mathcal{T}$	a set of edge types used in Chapter 5.
$q(\mathbf{z}_i)$	variational posterior distribution of vertex i used in Chapter 5.
$\log p(\cdot\|\cdot)$	log-likelihood of generation, or reconstruction term used in Chapter 5.
$p(\mathbf{z})$	predefined prior distribution used in Chapter 5.
$\mathcal{R}$	divergence metric used in Chapter 5.
l_d	length of document d used in Chapter 6.
$\mathcal{H}$	a set of pretrained word embeddings used in Chapter 6.
$\mathbf{h}_w$	pretrained word embedding of word w used in Chapter 6.
$\mathcal{D}_{\text{long}}$	the subset of long documents in corpus $\mathcal{D}$ used in Chapter 6.
$\mathcal{D}_{\text{short}}$	the subset of short documents in corpus $\mathcal{D}$ used in Chapter 6.
$\mathcal{T}_d$	the task of document d used in Chapter 6, i.e., generating observed words of d
$\mathcal{S}_d$	a set of support words of document d used in Chapter 6.
$\mathcal{Q}_d$	a set of query words of document d used in Chapter 6.
θ	a collection of parameters of encoder f_θ used in Chapter 6.
$\mathbf{m}_d$	missing semantics of document d used in Chapter 6.
μ	a collection of parameters of semantics prediction function g_μ used in Chapter 6.
$\mathcal{S}_{\mathcal{N}(d)}$	a set of support neighbors of document d used in Chapter 6.
$\mathcal{Q}_{\mathcal{N}(d)}$	a set of query neighbors of document d used in Chapter 6.

$\mathbf{t}_k$	topic embedding of topic k used in Chapter 7.
$\boldsymbol{\varepsilon}_i$	adjacency vector or neighbor distribution of document i used in Chapter 7, $\boldsymbol{\varepsilon}_i \in \mathbb{R}^N$
$\mathbf{s}_i$	structure embedding of document i used in Chapter 7.

Table 1: Notations.

Acknowledgments

Where there is a will, there is a way.

— Emperor Guangwu of Han

Recently, Singapore government announces that commuters no longer have to wear masks when taking public transport. I suddenly realize that when I just started my PhD journey in August 2018, I also did not have to wear masks. Time is flying. Five years later, it is February 2023 now, I will be graduating very soon.

It was in the afternoon of 31 July 2018, after eating chili fish cooked by my father as the farewell lunch, my father drived me and my mother to Changchun Longjia International Airport, Jilin, China. I told my parents "Come to visit Singapore when available! It is my treat!" My PhD journey just began after I landed in Singapore Changi Airport on 1 August 2018.

The first year of my PhD study was full of interesting moments. Singapore flyer took me to overview the splendid picture of night Singapore; Night Safari gave me the chance to know my "host family". I ate hamburger as my birthday dinner; I also celebrated Chinese New Year overseas for the first time. I finished six courses at school; I traveled to Thailand and Malaysia with my friends. More importantly, on one research meeting with my supervisor Prof. Hady W. Lauw in an evening of early September 2018, we came up with an amazing research idea. Lao Zi has a saying, "*A journey of a thousand miles begins with a single step*". That evening was my first step towards research world. I was really motivated to explore more.

I had been working hardly on the research idea until the end of the first year, and made a submission to AAAI-20, which eventually became the very first publication in my academic

career. My supervisor took me to travel to Hanoi, Vietnam in December 2019 and New York, US in February 2020 to present our research outcomes. I was grateful for the guidance and inspiration of my supervisor, as well as the encouragement and support of my parents. "*Read ten thousand books, travel ten thousand miles*". I was on the way to absorb more knowledge and meet more interesting people. These rewarding moments happened in my second year.

I was appointed as a teaching assistant in the third year, which excited me a lot because of my enthusiasm about teaching. I presented a tutorial about a computer vision project on IS712 Machine Learning course; I led a series of research seminars for young PhD students throughout the whole term. I also started to feel interested in swimming. My Vietnamese friends taught me swimming skills every week in SMU admin building. Additionally, the third year was also one of the most frustrating years I experienced until then. I got eight consecutive rejections for my research papers and did not make much progress. I started to doubt myself and lose confidence. I repeatedly watched the video recordings of Ma Long, a table tennis player who also experienced a long period of depression after his 2016 Rio Olympic champion and eventually peaked the table tennis world again, to encourage myself day and day. "*Nothing is impossible for a willing heart*". I appreciated myself who never gave up doing research. Eventually, I published three more first-authored papers by the end of the third year. However, I had not seen my parents in person for two years already.

After entering the fourth year, I was no longer a young PhD student and needed more dedication to research. I started to spend much more time doing research in the lab. I gradually became the first few students turning on the lights in the lab. I summarized the experiences of previous acceptances and rejections. I tried hard to come up with inspiring ideas and did comprehensive experiments. "*Diligence is the path up the mountain of knowledge*". I was grateful for my supervisor who always gave me valuable suggestions on how to move forward and my parents who always supported me no matter what difficulties I met.

In the evening of the day before my 27th birthday, I received a call from my mother telling

me the serious illness of my father. Without much consideration, I immediately flied back to my hometown. Have not seen my parents for three years, I had chance to stay with them for four days before my father had to go back to the hospital for further treatment. I had a great time with my parents during those four days. They enjoyed the dish I cooked for them; they were delighted to know I would graduate soon and continue studying as a postdoc. We took a family photo together with the warm sunshine in October. The sun still rises up in November, but unfortunately, my father's life suddenly stopped on 31 October 2022 forever. He will never have opportunity to come to Singapore and witness my PhD graduation; I will also never eat his lovingly prepared chili fish. Time is still ongoing. I will forever remember my father for his love and support all the time. These unforgettable moments happened in my early fifth year.

This is late February 2023. Five years ago, I received PhD offer from SMU also in late February and started to study with my supervisor Prof. Hady W. Lauw immediately in March. I would like to express sincere gratitude to him for his guidance, suggestions, and responsibility. His suggested research topic, document graph representation learning, five years ago eventually leads to fruitful research publications and becomes my dissertation title. His method to do research, coming up with a new angle and solving the research problem with a novel method, benefits me a lot and will continue making a positive impact in my future research. In addition, his positive life attitude and self discipline also teach me that life is not just about doing research, but also about equipping myself with social skills, such as communication methods and management ability. I have grown up in these five years not only in terms of learning research skills, but more importantly, in terms of behaving in a socially responsible way in the complex world. I would like to quote a saying by Confucius to express my appreciation to him, "*If in the morning I were to gain knowledge of the correct path in life, I would be able to die at sunset without regrets*".

I am grateful for Preferred.AI research group, who always keep an active life attitude and cheer me up whenever I meet difficulties. I quite enjoy doing exercises with them. I played table tennis with them on every Thursday in the first year; I enjoyed running with them around Marina

Bay in the second year; I learned how to swim from them in SMU admin building in the third year; I traveled with them to the US to attend academic conferences in the fourth year. Although coming from diverse countries, cultures, and backgrounds, we are always united by a common interest in both doing research and building a healthy lifestyle. I sincerely appreciate their help all the time, and hope all of them have a bright future.

My parents are the spiritual pillar throughout my whole life. No matter when I call them, they always immediately answer the phone and accompany me. My mother is a Chinese teacher at a local high school. She devotes her life to educating students and helping them pursue their dreamed universities. Motivated by her, I also dream to be a professor in the future to disseminate my knowledge to potential students and continue contributing to the research world. I therefore choose to become an instructor in the fifth year of PhD study to teach an undergraduate compulsory course, IS1702 Computational Thinking, at SMU. I hope I can formally become a professor in the near future to inherit my mother's occupation as well as her enthusiasm about education. My father was a production worker in a chemical company when he was young. But due to industrial revolution in 1990s in Jilin, China, he was laid off and had to look for other job opportunities. To support our family, he went to different places in China to earn salary. His steps never stopped even under the burning sun in Guangzhou and in the snowy days in Harbin and Liaoyuan. I seldomly had chance to stay with him. Most of the time, I could meet him only by video call over these years. He told me he was very proud of me when I achieved a good academic performance. Even though he will not have chance to witness my PhD graduation, I will definitely smile to the sky and tell him, "I achieve it!" I hope my mother has a healthy body; I hope my father enjoys his new life.

"*Where there is a will, there is a way.*" Life is always full of difficulties, but miracles also happen to everyone. I want to define the future of myself: working in universities to continue making contribution to education and doing further in-depth research. I can reciprocate the great kindness of SMU, as well as the world we live in. This is the end of my PhD journey, but is also

the beginning of my life. Finally, I would like to quote one more saying by Qu Yuan to finish my dissertation:

Long, long had been my road and far, far was the journey;

I would go up and down to seek my heart's desire.

Chapter 1

Introduction

1.1 Document Networks

Graph refers to a data type with vertices and links connecting them. Much of the data on the Web can be represented in a graph structure, ranging from social and biological to academic and Web page graphs, etc. By analyzing the structural topology of graph connectivity and the attributes within vertices, we are able to apply graph analysis in many real-world scenarios, text classification and information retrieval. To achieve these goals, one promising direction of recent deep learning techniques is called *graph representation learning*, i.e., learning the representation of each node by preserving its graph structure and attributes, which could in turn fuel tasks such as node classification, node clustering, and link prediction.

As a specific type of graph, text documents are usually interconnected in a graph structure. For example, Google Web pages are connected in a hyperlink network, academic papers constitute a citation network, Facebook user profiles are connected in a social network, news articles with similar tags are linked in a tag sharing network, etc. We call such data *document graph* or *document network*. Accurately understanding the main topics within documents can help in efficiently organizing the explosion of documents we encounter every day, such as Web page searching, academic paper indexing, friend recommendation based on user profiles, news arti-

cle classification, etc. Document embedding is a popular deep learning method to achieve this goal. We encode documents from high-dimensional vocabulary space into a low-dimensional embedding space and preserve their semantic similarity, so that documents describing similar content are embedded closely, while distinct documents are separated. As one important category of document embedding method, neural topic modeling represents each document as a low-dimensional topic distribution, and each topic is interpreted by a group of keywords.

Intuitively, two linked documents on the graph are likely to share similar topics, e.g., cited papers tend to discuss similar research problems. Graph connectivity reveals such document similarities, and modeling it could uncover meaningful insights. However, most previous topic models deal with the plain text within each document only, without considering graph connectivity among documents. To this end, we are motivated to propose neural topic models for networked documents to derive topic distributions for documents that preserve both text content and graph structure. By modeling both information, such unified representations would lead to more interpretable topics and better fulfill real-world applications, such as document searching, indexing, recommendation, and classification.

The first challenge is to model both textual content and graph connectivity. We will present the technical details of how to extend Auto-Encoders to achieve this goal at Chapter 3. In many real-world scenarios, documents are always associated with timestamps representing their creation time, e.g., academic papers have publication time, Web pages contain released time, etc. Modeling such temporal document networks would reveal how topics of documents evolve over the time and help us better understand the dynamic process of the corpus. We will explain the details of modeling time information at Chapter 4. We also observe that 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 the quality of document representations and the interpretability of a topic model, since documents by the same authors reveal similar semantics, and author-

ship 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. We will present the details of authorship modeling at Chapter 5. As another scenario, sometimes documents are quite short and contain only a few words, e.g., the titles of academic papers and news articles are sometimes the only observed content. It is challenging to accurately learn topics of those short texts on the graph due to limited text information. To alleviate such a problem, we respectively explore the effectiveness of meta-learning and pre-trained word embeddings for short text modeling on document networks at Chapter 6 and 7. Below we detail the challenges and the approaches we propose.

1.2 Challenges, Approaches, and Contributions

In this section, we point out the challenges of existing works, briefly describe the approaches we will adopt to improve current methods, and state the contributions of this dissertation. See Fig. 1.1 for an overview of this dissertation.

Auto-Encoder for Document Network Modeling

Since documents are connected in a network structure, the foremost research problem is to derive a neural topic model and preserve both textual content and network connectivity.

Challenges. We point out two challenges of existing works. *First*, most methods, such as PLSA [30] and LDA [6], focus on plain text but ignore the adjacency structure, which reveals the relationship across documents. Modeling the latter would uncover meaningful insight into latent semantics. *Second*, although there do exist topic models for networked documents, graphical models [10, 59] typically require a manual design of parameter estimation algorithms (e.g., variational inference and EM algorithm [4]), which limits the model flexibility.

Approaches. We are thus motivated to develop a neural topic model for networked doc-

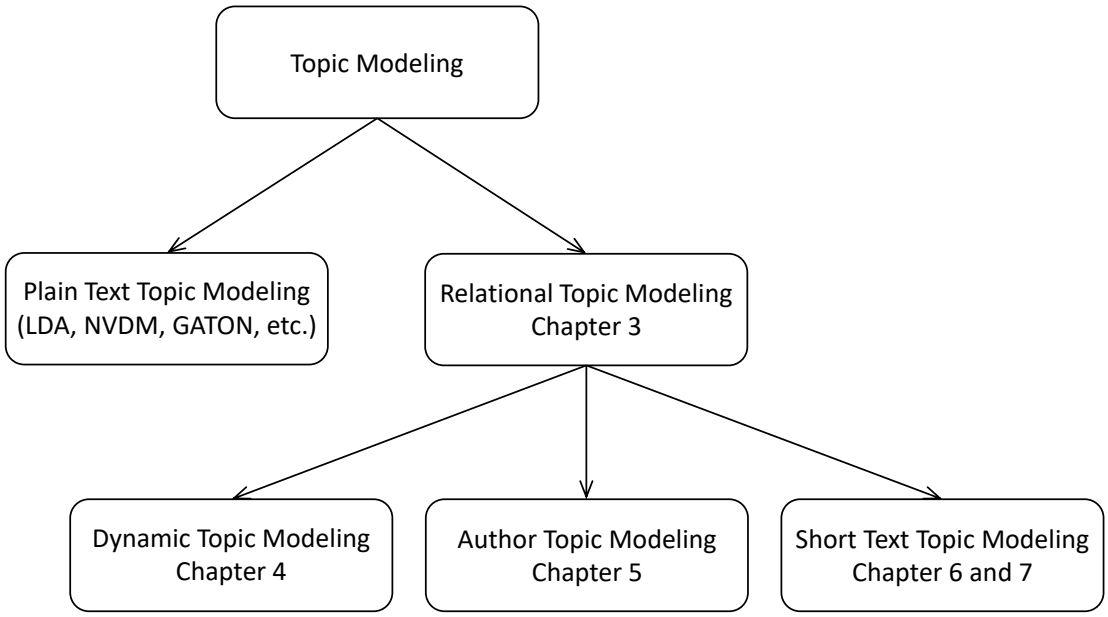


Figure 1.1: Overview of dissertation.

uments. We propose an approach called *Adjacent-Encoder*, whose key distinction is to also reconstruct the *neighbors* of the input document, in addition to the document itself. Hypothetically, this allows documents in a network to collaboratively learn from one another, such that close neighbors would have similar representations in the topic space. The realization of this principle leads to novel structures within the *Adjacent-Encoder* architecture.

Contributions. Correspondingly, for this document network modeling problem, we make the following contributions. *First*, we propose two novel architectures, *Adjacent-Encoder* and *Adjacent-Encoder-X*, as unsupervised topic models for document networks. *Second*, we systematically incorporate network structure in two ways, neighbor competition for topic propagation and neighbor reconstruction for semantic capture. Moreover, Adjacent-Encoder-X also investigates reconstruction of textual content and network structure. *Third*, we compare our models quantitatively and qualitatively against baselines of neural and graphical varieties on several evaluation metrics. *Fourth*, beyond showing improvements over comparable baselines, we investigate the complementarity and improved effectiveness of neighbor competition and reconstruction when combined with other architectural extensions such as denoising, contractive, and sparsity.

Our approaches and contributions lead to the following publication:

Topic Modeling on Document Networks with Adjacent-Encoder

Ce Zhang and Hady W. Lauw

Proceedings of the 34th AAAI Conference on Artificial Intelligence (**AAAI-20**)

Dynamic Topic Models for Temporal Document Networks

Where a document is associated with a timestamp representing its 'creation' time, the themes in a corpus may evolve over time. For examples, academic papers published over the years exhibit the development of research topics, chronologically released news articles present the change of storyline or events, etc. Capturing the dynamics in a sequentially organized corpus helps us better understand the evolution of topics. Dynamic topic model [5] is one such early attempt.

In many cases, text documents also link to one another in a network structure, e.g., newly published papers contain citations to existing papers, recent news articles hyperlink to older news. Modeling time could better preserve text semantics and network topology. Moreover, time also reveals the possibility of connection. Indicatively, newly published documents are likely to connect to recent neighbors rather than previous ones. However, existing topic models for document networks, e.g., RTM [10], focus on the static scenario and do not seek to preserve temporal evolution. As a result, it may predict a past link using links established in future time.

Challenges. We point out two challenges to existing methods in modeling dynamic document networks. *First*, models for networked documents [10] mainly focus on static networks without considering chronological events. Dynamic process showcases topic evolution and network generation over the time. By modeling it, we may better preserve text semantics and network topology. *Second*, most topic models [2, 10] preserve network structure by modeling its first-order neighborhood only, which could not make full use of network adjacency. The establishment of a link between two documents may be influenced by their common historical neighbors, which represent higher-order proximity. To tackle these challenges, we propose a

neural topic model for dynamic document networks that jointly preserves document dynamics and network adjacency.

Approaches. Optimal Transport (OT) [14] measures the distance between two probability distributions and has been successfully adopted by topic modeling in significantly improving topic coherence [33, 98, 118], but none has explored OT in a dynamic setting. In this work, we incorporate temporal information into OT and propose two neural topic models, NetDTM and NetDTM++, for **D**ynamic **T**opic **M**odeling on **Net**worked documents. Specifically, for NetDTM, in addition to the topic and word dimension, we add one more time dimension to OT and develop a Time-Aware Optimal Transport, which measures the probability of a link between two differently timestamped documents using their semantic distance. OT benefits our model by incorporating semantically related word embeddings in cost matrix. Besides the semantic-level modeling by NetDTM, we discover that the generation of a link is also influenced by the evolving topological structure of network. While NetDTM accounts for semantic modeling, we further propose NetDTM++ for network-level modeling, which designs a Temporal Point Process to capture the impact of network structure on the current link.

Contributions. We make the following contributions. *First*, we propose NetDTM and NetDTM++, which learn unified topic distributions to jointly preserve both document dynamics and network connectivity. *Second*, for NetDTM, by adding one more time dimension to optimal transport, we propose Time-Aware Optimal Transport to measure the distance between two differently timestamped documents for semantic modeling. *Third*, to model the effect of historical neighbors at the network level, for NetDTM++, we further encapsulate OT into a Temporal Point Process. *Fourth*, extensive experiments demonstrate the advantage of our models over baselines. We formulate above approaches and contributions into the following publication:

Dynamic Topic Models on Temporal Document Networks

Delvin Ce Zhang and Hady W. Lauw

Proceedings of the 39th International Conference on Machine Learning (**ICML-22**)

Graph Neural Networks for Authors Topic Modeling

We observe that 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 the quality of document representations and the interpretability of a 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 [6], infer topics based on plain text only, without auxiliary *authorship* or *venues*.

Challenges. Most existing graph neural networks for text embedding, e.g., TextGCN [103], lack topic modeling, leading to uninterpretable representations. Although there exist a few studies [20, 91] 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 [6] and the recent GATON [100], 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 [73] and ACT [78], 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 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. The proposed method leads to the following publication:

Variational Graph Author Topic Modeling

Delvin Ce Zhang and Hady W. Lauw

Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (**KDD-22**)

Meta-Learning for Short Text Topic Modeling

The quality of topic distribution of each document depends on sufficient word co-occurrences. However, many real-world corpora contain documents of *variable lengths*. Academic papers vary from journal manuscripts to conference papers to extended abstracts. News articles could be headlines, short or full articles, or detailed commentaries. Despite variable lengths (with different degrees of sufficiency of word co-occurrences), existing works, e.g., ProdLDA [76] and GATON [100], treat documents uniformly, resulting in inferior topic quality for short texts.

Challenges. Most existing topic models optimize the learning process by averaging the gen-

erative losses of different documents, without paying special attention to semantically limited short texts. A few studies, e.g., OTLDA [33], take weighted summation of losses based on document lengths, they compute the weights by dividing the length of each document by the length of the whole corpus, which further deemphasizes the importance of short text modeling. Thus, we seek to improve short text topic modeling within a *variable-length* corpus, without hurting topic quality of long documents.

To mitigate the scarcity of word co-occurrences in short texts, some works leverage auxiliary knowledge to enhance topic modeling. ETM [19] exploits pretrained word embeddings [61, 68] to capture word similarities. RTM [10] constructs a document network, e.g., paper citation network, to aggregate topics of connected documents. However, they rely on the availability of auxiliary data.

Approach. We propose **M**eta-**C**omplement **T**opic **M**odel (MCTM). Since a corpus contains variable lengths of documents, we are motivated to learn the transferable semantic knowledge on long documents, and complement the semantics-scarce short texts and enhance the latter's topic distributions. Since meta-learning [23] is emerging to improve model performance with few labeled observations, but no one explores its design in topic modeling. We are thus motivated to integrate it and optimize the proposed MCTM by a meta-learning objective.

Orthogonal to existing works relying on auxiliary data, our framework is self-contained, assuming only in-corpus information, which offers a new direction to improve short text topic modeling. When auxiliary data are available, our framework can be further improved by flexibly incorporating them. In particular, when incorporating document network structure, we discover that document degrees also exhibit a similar long-tail distribution, i.e., some structure-abundant documents link to sufficient neighbors as auxiliary, while others contain scarce links.

Contributions. Our contributions are as follows. *First*, we propose MCTM, which learns how to complement textual semantics by semantic knowledge transfer. *Second*, we derive two alternatives to implement missing semantics prediction function to capture document similari-

ties. *Third*, although agnostic to auxiliary data, MCTM can also flexibly integrate them to further improve the performance. We demonstrate our adaptability by modeling pretrained word embeddings and document networks. For the latter, we extend MCTM to further complement structural semantics. *Fourth*, extensive experiments verify the effectiveness of MCTM. We organize the proposed research idea into the following publication:

Meta-Complementing the Semantics of Short Texts in Neural Topic Models

Delvin Ce Zhang and Hady W. Lauw

Proceedings of the 36th Conference on Neural Information Processing Systems (**NeurIPS-22**)

Optimal Transport Barycenter for Short Text Topic Modeling

Topic modeling replies on word co-occurrences within documents to learn effective topic distributions. Words that frequently co-occur with each other tend to reveal consistent topics, and different topics represent distinct word co-occurrence patterns. However, when documents are quite short with only a few words, accurately discovering latent topics becomes extremely challenging. For example, the title of Google Web pages and news articles usually contains less than 20 words; the abstract of academic papers usually has less than 100 words.

Therefore, the **challenge** of existing works is that models for networked documents (e.g., NRTM [2]) tend to deteriorate for shorter documents with fewer word co-occurrences, resulting in less interpretable topics and worse task performances.

Approaches. Pre-trained word embeddings, such as word2vec [61] and GloVe [68], on an external large corpus (Wikipedia and Google pages) preserve semantic similarity and word co-occurrence patterns. Words that frequently co-occur or present similar context are embedded closely. Even though two words do not co-occur within the same document, their similar word embeddings would allow one topic to activate both of them simultaneously. Thus, the co-occurrence pattern is still captured by external knowledge, and topic quality can be improved. By incorporating pre-trained word embeddings as auxiliary information into topic modeling, we

are able to alleviate the short text problem.

Recently, Optimal Transport (OT) [14] has been employed on machine learning problems and achieved promising performance, including generative [33] and neural topic modeling [118]. But no existing such method explores the modeling of document connectivity. Therefore, for network structure modeling, we are motivated to develop neural topic model built on the theory of *Optimal Transport Barycenter* [15]. Unlike conventional topic models that leverage a document's topic distribution to generate its own observed content, OT barycenter naturally allows the topic distribution of a document to generate the content of not only itself, but also its linked neighbors. Such mechanism matches the intuition that if two linked documents share similar topics, it is possible to use topic distribution of one document to generate the content of the other, even though their observed texts are different.

For semantic interpretability, we extend the cost matrix of optimal transport and incorporate *pre-trained word embeddings*, which lead to interpretable topics even when text documents are short with a few word co-occurrences. Since Dirichlet prior distribution in LDA [6] successfully improves topic quality, we are also motivated to investigate Dirichlet as an *optimal transport prior distribution* to further boost topic interpretability.

In this chapter, we extend Variational Graph Auto-Encoder (VGAE) [40], a specific variant of GNNs, and integrate these modeling approaches into a unified framework, named **DBN** for **D**irichlet Optimal Transport **B**arycenter for Document **N**etworks, which captures document network connectivity, and the learned topic distributions enjoy semantic interpretability.

Contributions. Our contributions are as follows. *First*, we propose DBN, a VGAE topic model that unifies document network modeling and semantic interpretability into a joint graph neural network framework. *Second*, to model network structure, we propose Optimal Transport Barycenter, which induces barycentric topic distributions of documents by generating observed content of network neighbors. *Third*, for semantic interpretability, we extend the cost matrix of optimal transport by incorporating pre-trained semantically related word embeddings. We further

propose Dirichlet distribution as an optimal transport prior to boost topic quality. We organize the above approaches and contributions into the following submission:

Topic Modeling on Document Networks with Dirichlet Optimal Transport Barycenter

Delvin Ce Zhang and Hady W. Lauw

Being reviewed

Chapter 2

Related Work

2.1 Graph Representation Learning

Homogeneous Graphs. Homogeneous networks are those with one single type of vertices and links. DeepWalk [69] generates random walk on the network as corpus and applies skip-gram model to train the nodes. Node2vec [27] extends DeepWalk by simulating biased random walk to explore diverse neighborhoods. LINE [77] learns node representations by preserving first- and second-order proximities. GraRep [9] generalizes LINE to incorporate higher-order proximities, but may not scale efficiently to very large networks. There are also some methods focusing on temporal graph embedding [54, 123].

Meanwhile, graph neural networks represent an important class of models for graph-structured data. GCN [39] extends CNN [43] and leverages convolution operation on graphs to aggregate neighboring information to learn node embeddings. GAT [83]) designs multi-head attention mechanism to evaluate different importance of neighbors. GraphSAGE [29] proposes several aggregators to support inductive node representation learning. Recent works exploit more structural information on graphs, such as graph isomorphism [99] and node positions [105]. Variational Graph Auto-Encoder (VGAE) [40] extends VAE [38] where Graph Convolutional Network (GCN) [39] is the vertex encoder. ARVGA [65] improves VGAE by adversarial training. DG-

VAE [49] replaces Gaussian prior with Dirichlet. Graphite [28] extends the decoder of VGAE by an iterative graph refinement strategy. CGVAE [51] investigates the application in chemistry. MLHNE [110] constructs a multi-layered graph. TGAT [16] models dynamic process of a temporal graph.

Graph neural networks (GNNs) learning text embeddings are also proposed. TextGCN [103] designs a document-word graph and applies GCN to learn text embeddings for text classification. It is further extended by TensorGCN [39] to incorporate sequential, syntactic, and semantic word relationships. TextING [116] and HyperGAT [20] support inductive text embedding by designing graph for each individual document. There are also GNNs designed for topic modeling on graphs. GATON [100] applies GCN on a bipartite graph for topic modeling. GraphBTM [121] improves biterm topic model using graphs. GTNN [94] extends GATON for a two-layered network. TVGAE [95] extends VGAE [40] for topic modeling. Similar works also include DHTG [91], GRTM [93], LANTM [90], GNCTM [96], MCTM [109], etc.

Heterogeneous Graphs. Some heterogeneous network models leverage meta-path-based random walks to capture network semantics, such as Metapath2vec [21] and HIN2vec [25]. The applications of meta-path-based models (e.g., recommender systems) are also widely studied [74]. Some of them simulate meta-paths of specified schemes on each network to preserve complex semantics. There also exist some methods that do not require specific meta-paths, such as HeGAN [31], which utilizes GAN [26] to generate fake nodes to train discriminator. More recently, Graph Neural Networks have been successfully applied to attributed heterogeneous networks with satisfactory results [89].

Multi-layered networks, as a set of interdependent network layers, are a another category of heterogeneous networks. They appear in real-world scenarios including recommender and academic systems, cross-platform social networks, etc. Previous works focus on cross-layer links inference [11, 12] and network ranking [64]. MANE [48] studies representation learning on multi-layered networks by seeking low-dimensional node embeddings by modeling each intra-

and cross-layer links. MLHNE [110] extends MANE by modeling higher-order proximities.

2.2 Neural Topic Modeling

Auto-Encoders. There are architectural variants to Auto-Encoder that have been shown to improve the performance of topic modeling. Denoising Auto-Encoder (DAE) [84] adds random noise to the input document and reconstructs its original content to learn useful patterns while avoiding overfitting. Contractive Auto-Encoder (CAE) [72] introduces the Frobenius norm of Jacobian matrix to the loss function for regularization. K-Sparse Auto-Encoder (KSAE) [55] and K-Competitive Auto-Encoder (KATE) [13] force topics to be sparse by keeping the values of only k hidden neurons and zeroing others. Variational Auto-Encoder (VAE) [38] makes use of variational inference to learn topics in a generative approach. Motivated by the powerful framework of VAE, subsequent research is conducted for the development of neural topic modeling. ProdLDA [76] uses product of experts to generate words in contrast to LDA's mixture assumption. DVAE [8] proposes to use Dirichlet distribution as prior to increase topic sparsity, WHAI [115] applies Gamma distribution, and NVDM [60] adopts Gaussian distribution. GTM [119] explores higher-order word co-occurrences for topic modeling.

Word Embedding Based Topic Models. Short documents or short texts generally have fewer words than long documents. This makes topic modeling more challenging, since the given texts are too limited to infer high-quality topics. To alleviate the sparsity in short documents, some topic models incorporate pre-trained word embeddings, such as word2vec [61] and GloVe [68], to improve semantic learning. Previously, such methods are mainly the extensions of LDA [6]. They use word embeddings and topic embeddings to define topic-word distribution. GLDA [17] generates each word using a Gaussian distribution defined by topic and word embeddings. LCTM [32] reveals topics by the co-occurrence of latent concepts. GPUMM [47] improves the modeling by a Dirichlet Multinomial Mixture model. MetaLDA [117] converts word embeddings to binary encoding format and treats them as a general meta information. In addition to the

graphical generative models, neural models based on VAE are also developed. The recent neural model ETM [19] injects word embeddings into the decoding of VAE.

Optimal Transport Based Topic Models. There are only a few works that develop generative and neural topic models on optimal transport [14]. To our knowledge, the recently proposed NSTM [118] is the only neural model, which minimizes topic distribution and word distribution by OT distance. Again, it does not model document adjacency. DWL [98] and OTLDA [33] are generative models. DWL is mainly designed to embed codes of international classification of diseases, while our models are for general topic modeling. Both DWL and OTLDA apply OT in the word space only, while ours integrate OT in topic, word, and structure spaces. Other models [63, 66, 80] leverage OT to measure the distance between the generated distribution and the true distribution. All above methods model plain text only.

Dynamic Topic Models. For documents with timestamps representing their publication time, dynamic models leverage time information to improve topic modeling. The pioneering method is DTM [5], which uses a Markov chain [4] to capture semantic evolution. cDTM [88] improves DTM by Brownian motion for continuous time modeling. DETM [18] is a neural model that injects word embeddings into the decoding process. MDTM [34] allows online update of its dynamic parameters. Others [3, 35] speed up the inference process by sampling methods. While these models incorporate time information for dynamic modeling, they ignore the network adjacency across documents. NetDTM [113] is the first to incorporate both document network connectivity and time information for topic modeling.

Author Topic Models. Author Topic Model (ATM) [73] derives topics for authors. ACT [78] improves ATM by modeling venues. CAT [81] further models paper citations. They do not infer topics for documents. CNTM [50] infers topics for both documents and authors, but fails to consider venues. VGATM [114] is the first to model authors and venues, and meanwhile learn topic distributions for documents.

Supervised Topic Models. Supervised and semi-supervised topic models are those methods

that embed both textual content and document labels and produce label-dependent topic distributions. sLDA [57] is designed with a regression component and supervised by numerical values. Other models, such as DiscLDA [41], modify topic distributions for categorical label supervision. LabeledLDA [70] is proposed for multi-label documents, while PLDA [71] is for partially labeled documents. MedLDA [120] integrates the max-margin concept into supervised topic models. These mentioned methods are based on graphical models. SemiVAE [37] and MVAE [92] are based on Auto-Encoder, a neural topic model.

Document Network Models. There are some models for networked documents. NetPLSA [59], RTM [10], and PLANE [42] are graphical models. They use topic distributions of two documents to generate the link between them. NRTM [2] is a neural model that applies VAE to encode documents and multi-layer perceptron [4] to predict the links. Adjacent-Encoder [107] captures network structure by neighbor reconstruction. They model the effect of network, but ignore the dynamic process of network generation. SemiVN [111] explores supervised learning on document networks.

Others. Some topic models discover topics of documents based on document comparison. CompareLDA [79] is designed for single-aspect pairwise document comparisons, while MALIC [112] is for multi-aspect listwise document comparisons. Some other models [67, 106] aim at discovering common topics across multiple collections of documents and collection-specific topics.

Meta-Learning. Here, we also briefly review meta-learning works. One category is metric-based, e.g., ProtoNet [75] and MatchingNet [86], which learn a metric function over tasks. Gradient-based meta-learning works optimize model parameters for quick adaptation to new tasks [23, 24, 44, 87, 101, 102, 104].

Chapter 3

Auto-Encoder for Document Network Modeling

3.1 Introduction

In this chapter, we investigate neural topic models not for plain-text documents per se, but for networked documents. In addition to textual content, oftentimes documents link to one another in a network structure. For example, academic papers form a citation network, Web pages form a hyperlink network. Many previous works on topic modeling focus on textual content of documents; some do incorporate the network structure to jointly learn representations, such as RTM [10]. To this end, novel approaches to unsupervised topic modeling for document networks are germane, because of their importance and wide applicability.

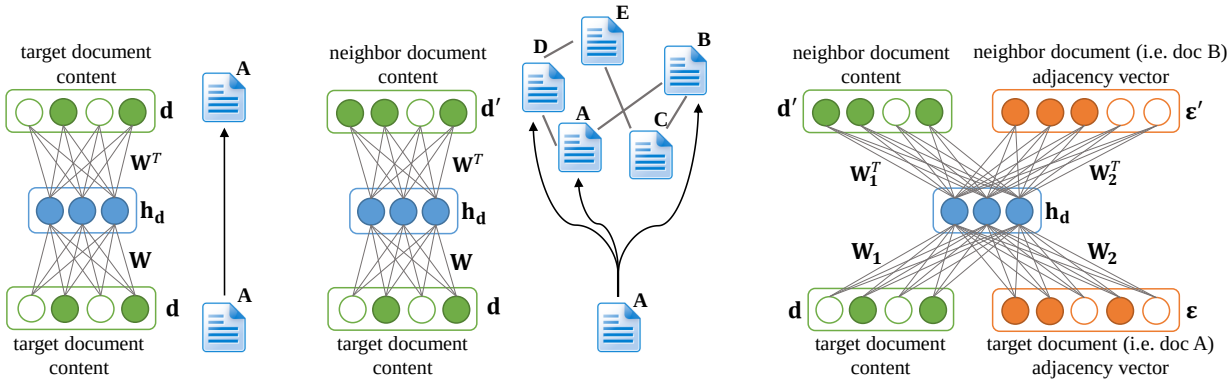
Definition 3.1.1 (Document Network)**.** *Let $\mathcal{G} = (\mathcal{D}, \mathcal{E})$ be a given document network. $\mathcal{D} = \{\boldsymbol{d}_i\}_{i=1}^N$ is a corpus of documents, where each document $\boldsymbol{d}_i \in \mathbb{R}^{|V|}$ is a vector in the vocabulary space $\mathcal{V}$. N is the total number of documents in the corpus. In turn, the adjacency matrix $\mathcal{E} \in \mathbb{R}^{N \times N}$ is a 0-1 matrix where $\varepsilon_{ij} = 1$ indicates document i links to j, and $\varepsilon_{ij} = 0$ otherwise. Here we model an undirected network, i.e., $\varepsilon_{ij} = \varepsilon_{ji}$ and $\mathcal{E} = \mathcal{E}^\top$, though the proposed models could generalize to directed networks as well. We would use edge and link interchangeably. For*

a document i, its neighbors are those directly linked to i. For simplicity, we use $\mathcal{N}(i)$ to denote i's neighbor set. The definition of neighborhood here is reflexive, i.e., we also regard i as its own neighbor, $i \in \mathcal{N}(i)$ and $\varepsilon_{ii} = 1$.

Given $\mathcal{G}$ as input, our aim is to embed documents in $\mathcal{G}$ to low-dimensional topic distributions, which preserve both textual content $\mathcal{D}$ and network structure $\mathcal{E}$. Recent neural topic models are based on the traditional *Auto-Encoder* family, which naturally embodies the notion of a topic model, by learning the association between documents and topics (hidden neurons), as well as topics and words. However, in seeking to reconstruct the input document, it would model each document independently and disregard the network structure in $\mathcal{G}$. To deal with networked documents, we propose an approach called *Adjacent-Encoder*, whose key distinction is to also reconstruct the *neighbors* of the input document, in addition to the document itself.

- **Neighbor Competition**: Neighbors contribute information differentially. In the encoding phase, we evaluate attentions between the target document and its neighbors. In turn, neighbors propagate topics to the target document.
- **Neighbor Reconstruction**: In the decoding phase, the target document reconstructs the contents of its adjacent neighbors. This increases the robustness and invariance of topic representations with respect to output documents, while also incorporating the neighborhood structure without additional parameters over those of Auto-Encoders.

Beyond reconstructing the content of neighbors, it is feasible to reconstruct their neighborhood structure as well. This factors in higher-order proximities by modeling the adjacency matrix explicitly. We realize this in an extension *Adjacent-Encoder-X*, which *jointly* embeds content and network structure in a unified manner.



(a) Auto-Encoder (b) Adjacent-Encoder (c) Adjacent-Encoder-X

Figure 3.1: Comparison among Auto-Encoder, Adjacent-Encoder, and Adjacent-Encoder-X.

3.2 Background

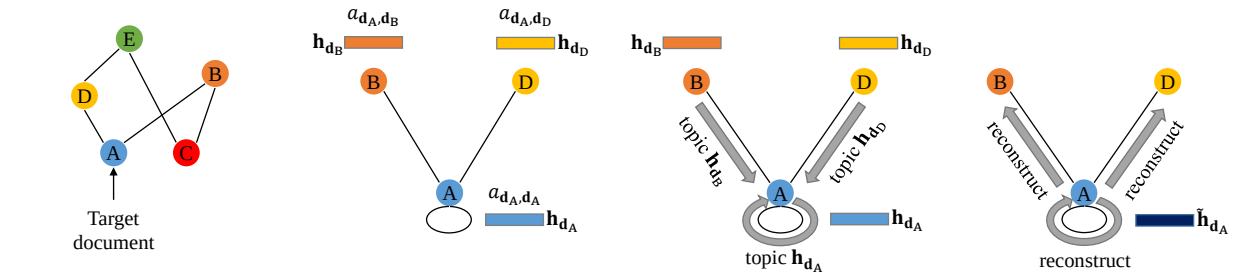
We briefly review Auto-Encoder to make our contrast clear. With activation function f, we learn hidden representation $\mathbf{z}_i$ for the input document i at the encoder: $\mathbf{z}_i = f(\mathbf{W}\mathbf{d}_i + \mathbf{b})$. The decoder reconstructs the original content of the input document by $\hat{\mathbf{d}}_i = f'(\mathbf{W}'\mathbf{z}_i + \mathbf{c})$. Here $\mathbf{b} \in \mathbb{R}^K$ and $\mathbf{c} \in \mathbb{R}^{|\mathcal{V}|}$ are biases, $\mathbf{W} \in \mathbb{R}^{K \times |\mathcal{V}|}$ and $\mathbf{W}' \in \mathbb{R}^{|\mathcal{V}| \times K}$ are encoder and decoder parameters. Typically we use weight tying ($\mathbf{W}' = \mathbf{W}^\top$) as regularization. $\mathcal{V}$ is vocabulary, and K is the number of hidden neurons, or the number of topics. By minimizing the reconstruction error, we obtain $\mathbf{z}_i$ as topic representations.

3.3 Model Architecture and Analysis

In this section, we describe the technical details of our proposed models, *Adjacent-Encoder* and *Adjacent-Encoder-X*.

3.3.1 Adjacent-Encoder

Fig. 3.1 contrasts our proposed models *Adjacent-Encoder* (Fig. 3.1(b)) and *Adjacent-Encoder-X* (Fig. 3.1(c)) with traditional Auto-Encoder (Fig. 3.1(a)). Here we describe *Adjacent-Encoder* by



(a) Input document network (b) Neighbor competition (c) Topic propagation (d) Neighbor reconstruction

Figure 3.2: Illustration of neighbor competition, topic propagation, and neighbor reconstruction.

highlighting its constituent structures, and defer *Adjacent-Encoder-X* to the next section.

As a running example, we assume a toy network of 5 documents $\{A, B, C, D, E\}$ as in Fig. 3.1(b). The key principle behind *Adjacent-Encoder* is to have a target document, say A, reconstruct itself and its adjacent neighbors, say B and D. This manifests via the mechanisms illustrated in Fig. 3.2.

Neighbor Competition. The first is to allow competition among documents to assess relative importance among neighbors. As in [13], we represent each input document as a log-normalized word count vector $\mathbf{d} \in \mathbb{R}^{|\mathcal{V}|}$, i.e., each dimension is $d_w = \frac{\log(1+n_w)}{\max_{w \in \mathcal{V}} \log(1+n_w)}$ where n_w is the count of word w in $\mathbf{d}$. For a target document i, we learn its hidden vector $\mathbf{z}_i$ at the feedforward phase by $\mathbf{z}_i = tanh(\mathbf{W}\mathbf{d}_i + \mathbf{b})$. The attention coefficients between i's neighbors and itself are a_{ij} as shown in Fig. 3.2(b).

$$\tilde{a}_{ij} = \mathbf{z}_i^\top \mathbf{z}_j, \qquad a_{ij} = \frac{\exp(\tilde{a}_{ij})}{\sum_{j' \in \mathcal{N}(i)} \exp(\tilde{a}_{ij'})}. \tag{3.1}$$

$j \in \mathcal{N}(i)$ is a neighbor of i. The attention measures relative importance among i's neighbors.

Links among documents indicate a shared relationship. Thus we propagate the topics of neighbors to the target document i, which, in turn, is also a neighbor of other documents, thereby propagating topics even further. We allow topics to flow through neighbors across the network, so that documents collaboratively learn from one another. This procedure is driven by the following

transformation, which is also illustrated by Fig. 3.2(c).

$$\tilde{\mathbf{z}}_i = \sum_{j \in \mathcal{N}(i)} a_{ij} \mathbf{z}_j. \tag{3.2}$$

Neighbor Reconstruction. The content of a document is the observed reflection of its internal topics. Since we know linked documents are likely to share similar topics, we could use topic representation $\tilde{\mathbf{z}}_i$ of a target document to reconstruct the contents of its adjacent neighbors in a "1-to-N" reconstruction manner. We adopt $sigmoid$ as the output activation function $\sigma(x) = \frac{1}{1+\exp(-x)}$, and weight tying is used for regularization.

$$\hat{\mathbf{d}}_i = \sigma(\mathbf{W}^T \tilde{\mathbf{z}}_i + \mathbf{c}) \tag{3.3}$$

where $\hat{\mathbf{d}}_i$ is the reconstruction document. We use binary cross-entropy as the loss function.

$$l(\mathbf{d}_j, \hat{\mathbf{d}}_i) = -\sum_{w \in \mathcal{V}} [d_{j,w} \log(\hat{d}_{i,w}) + (1 - d_{j,w}) \log(1 - \hat{d}_{i,w})]. \tag{3.4}$$

Again, $j \in \mathcal{N}(i)$ is one of the adjacent neighbors. We have each target document i reconstruct each of its neighbors, as illustrated in Fig. 3.2(d). Each document in the network takes turn as the target document. We repeat the learning process for unsupervised training until convergence.

Reconstructing adjacent neighbors is somewhat related to Denoising Auto-Encoder (DAE), which reconstructs a document from a noisy version of itself. Instead of noise at the input layer, our models have a target document reconstruct adjacent neighbors, which in a way serves as "noise" at the output layer. In our case, the "noise" is naturally introduced by the network, instead of being a random and artificial addition to documents. The reconstruction of neighbors allows our models to capture the case where two documents are different in observed content, but consistent in terms of the internal topics, thus the learned topics can preserve document semantics well. Furthermore, it also increases the robustness and invariance of the learned topics

w.r.t. output documents.

Inference. Once our models have been trained, we simply encode each testing document $\mathbf{d}'$ by $\mathbf{z}' = tanh(\mathbf{W}\mathbf{d}' + \mathbf{b})$. Here $\mathbf{z}'$ is the topic representation of the testing document, which preserves information of both text and network structure.

3.3.2 Adjacent-Encoder-X

The previously described *Adjacent-Encoder* models network structure implicitly. In this section we propose an improved framework, *Adjacent-Encoder-X*, which models network structure explicitly. The distinction of these two is illustrated in Fig. 3.1. The name is inspired by the 'X' structure of dual observations of textual content and adjacency vector.

Neighbor Competition. Adjacency matrix $\mathcal{E}$ represents the network structure. The i^{th} row (or column) $\boldsymbol{\varepsilon}_i$ represents the neighborhood relationship of i^{th} document. If two documents have many common neighbors, their corresponding adjacency vectors are similar. Intuitively, the more common neighbors two documents have, the more likely they share similar topics. Two academic papers may share similar topics if both cite many of the same papers. Web pages may be of the same category if they link to common Web pages.

Hence, we treat the adjacency vector $\boldsymbol{\varepsilon}_i$ as another input in addition to the textual content $\mathbf{d}_i$. The hidden vector of the feedforward phase can be computed by $\mathbf{z}_i = tanh(\mathbf{W}_1\mathbf{d}_i + \mathbf{W}_2\boldsymbol{\varepsilon}_i + \mathbf{b})$. Here $\mathbf{W}_1 \in \mathbb{R}^{K\times|\mathcal{V}|}$ and $\mathbf{W}_2 \in \mathbb{R}^{K\times N}$ are parameters for textual content and adjacency vector respectively. $\mathbf{b} \in \mathbb{R}^K$ is bias. N is the total number of documents.

The remaining process of neighbor competition for *Adjacent-Encoder-X* is similar to *Adjacent-Encoder*. Thereafter, we obtain the aggregate hidden vector $\tilde{\mathbf{z}}_i$.

Neighbor Reconstruction. We still have each target document reconstruct its adjacent neighbors, but now in terms of both textual content and adjacency vector.

$$\hat{\mathbf{d}}_i = \sigma(\mathbf{W}_1^\top \tilde{\mathbf{z}}_i + \mathbf{c}_1), \quad \hat{\boldsymbol{\varepsilon}}_i = \sigma(\mathbf{W}_2^\top \tilde{\mathbf{z}}_i + \mathbf{c}_2) \tag{3.5}$$

Model	#Parameters
Adjacent-Encoder	$K\|\mathcal{V}\| + K + \|\mathcal{V}\|$
Adjacent-Encoder-X	$K\|\mathcal{V}\| + KN + K + \|\mathcal{V}\| + N$
AE, DAE, CAE, KSAE, KATE	$K\|\mathcal{V}\| + K + \|\mathcal{V}\|$
VAE	$3K\|\mathcal{V}\| + 2K + \|\mathcal{V}\|$

Table 3.1: Number of parameters.

Name	#Labels	#Documents	#Links	Vocabulary
DS	9	570	1,336	3,085
HA	6	223	515	2,073
ML	7	1,980	5,748	4,431
PL	9	1,553	4,851	41,05

Table 3.2: Dataset statistics.

where weight tying is used, and $\mathbf{c}_1 \in \mathbb{R}^{|\mathcal{V}|}$ and $\mathbf{c}_2 \in \mathbb{R}^N$ are biases. The loss function for textual content is given by (Eq. 3.4), and the loss function for adjacency vector is given below.

$$l(\boldsymbol{\varepsilon}_j, \hat{\boldsymbol{\varepsilon}}_i) = -\sum_{n=1}^{N} [\varepsilon_{j,n} \log(\hat{\varepsilon}_{i,n}) + (1 - \varepsilon_{j,n}) \log(1 - \hat{\varepsilon}_{i,n})]. \tag{3.6}$$

$\boldsymbol{\varepsilon}_j$ represents the adjacency vector of i's neighbors. Each target document reconstructs its neighbors in these two aspects, generating the total loss function $l = l(\mathbf{d}_j, \hat{\mathbf{d}}_i) + l(\boldsymbol{\varepsilon}_j, \hat{\boldsymbol{\varepsilon}}_i)$.

Inference. Upon convergence we encode a testing document $\mathbf{d}'$ by $\mathbf{z}' = tanh(\mathbf{W}_1\mathbf{d}' + \mathbf{W}_2\boldsymbol{\varepsilon}' + \mathbf{b})$. $\mathbf{z}'$ is the topic representation encompassing text content and network structure.

3.3.3 Complexity Analysis

Model Complexity. Table 3.1 lists the parameter counts for our models and the Auto-Encoder family. For *Adjacent-Encoder*, we set $\mathbf{W}' = \mathbf{W}^\top$ of dimensionality $K|\mathcal{V}|$. The only other parameters are biases of size K and $|\mathcal{V}|$. Note that compared to other Auto-Encoder models (AE, DAE, CAE, KSAE, KATE), *Adjacent-Encoder* does not add extra parameters as it models the network structure implicitly. For *Adjacent-Encoder-X*, because the adjacency matrix is another

input in addition to the content, the number of parameters is now $K|\mathcal{V}| + KN + N + K + |\mathcal{V}|$.

Computational Complexity. We use F to denote the number of input features ($|\mathcal{V}|$ and $|\mathcal{V}| + N$ respectively for *Adjacent-Encoder* and *Adjacent-Encoder-X*). The feedforward complexity for each target document is $\mathcal{O}(KF)$. For neighbor competition and topic propagation, let $\deg_{\max}$ denote the maximum number of neighbors in the network. The complexity of each target document is $\mathcal{O}(K \deg_{\max})$. For neighbor reconstruction, we reconstruct all adjacent neighbors, thus we have $\mathcal{O}(KF \deg_{\max})$. Putting all three components together, for each target document, we obtain $\mathcal{O}(KF + K \deg_{\max} + KF \deg_{\max})$ for the overall model. In comparison, Auto-Encoders usually have $\mathcal{O}(KF)$ complexity. Although our models bring additional complexity, we are able to further incorporate document network structure to improve the performance. Finally, since the main emphasis of this work is model effectiveness, but not running efficiency, we consider speeding up the training process as a future work.

3.4 Experiments

Our experimental objective is to validate the quality of topics learned by our models on evaluative tasks such as document classification, document clustering, link prediction, etc.

3.4.1 Setup

Datasets. Cora [58] is a public collection of papers and their citations. Each document is an abstract. Two documents are linked by an undirected edge if one cites the other. Following [122], we extract four independent datasets: Data Structure (DS), Hardware and Architecture (HA), Machine Learning (ML), and Programming Language (PL). Each dataset is organized into categories, which we treat as class labels (not used in learning, only evaluation). Table 3.2 presents their statistics.

Baselines. We compare our models against several categories of baseline models below.

- **Auto-Encoders**: Since our models are encoders, the most appropriate baselines are of the Auto-Encoder family, i.e., AE, DAE [84], CAE [72], VAE [38], KSAE [55], and the state-of-the-art topic model KATE [13]. As they encode only the document content, through this comparison we validate the efficacy of jointly learning content and network structure.
- **Generative topic models**: Another family of topic models are based on the generative approach. We compare to those that incorporate document content and network structure concurrently, such as RTM [10], PLANE [42], and the recent NRTM [2]. We also include ProdLDA [76], a recent topic model that still encodes each document independently.
- **Graph embedding**: Recently there are some models making use of Auto-Encoder for unsupervised graph representation learning. Strictly speaking, they are not topic models, nor baseline. For completeness, we include a comparison to VGAE [40].

Training Details. Following [2, 13], the activation functions for AE, DAE, CAE, KSAE, and NRTM are $sigmoid$, while those for VAE and KATE are $tanh$ (hidden) and $sigmoid$ (output) respectively. We use validation set to choose the best hyperparameters. DAE with Gaussian noise of 0.25 std.dev. outperforms other kinds of noise. We choose 2 and 0.01 as Dirichlet hyperparameter for RTM and PLANE. For KSAE and KATE, we set the number of nonzero hidden neurons, k, to 4, 8, 16, 32, and 52 when the number of topics is 16, 32, 64, 128, and 256, respectively. Each result is an average of 10 independent runs.

Transductive vs. Inductive Learning. There are two scenarios in which we can apply the models. In the transductive setting, the objective is to derive topic representations of the documents already in the corpus. In this case, all documents in the corpus are present during training. Conversely, in the inductive setting, the objective is to generalize beyond the training corpus to unseen data, which we simulate by keeping a random subset of 80% documents for training (out of which we further randomly split 10% documents for validation) and the remaining 20% for testing. As both are feasible scenarios, we discuss our experiments under each setting.

	Transductive Learning											
Model	Document Classification				Document Clustering				Link Prediction			
	DS	HA	ML	PL	DS	HA	ML	PL	DS	HA	ML	PL
Adjacent-Encoder	0.739	0.842	**0.864**	0.772	**0.470**	0.540	0.564	0.388	**0.396**	**0.331**	0.226	0.237
Adjacent-Encoder-X	**0.744**	**0.846**	0.857	**0.780**	0.445	**0.548**	**0.571**	**0.392**	0.374	0.326	**0.251**	**0.271**
AE	0.558	0.688	0.739	0.616	0.250	0.315	0.368	0.230	0.144	0.195	0.107	0.102
DAE	0.656	0.799	0.790	0.694	0.372	0.409	0.441	0.278	0.204	0.296	0.121	0.147
CAE	0.558	0.685	0.741	0.620	0.261	0.309	0.371	0.228	0.145	0.188	0.108	0.103
VAE	0.652	0.789	0.796	0.679	0.356	0.394	0.447	0.286	0.193	0.283	0.122	0.135
KSAE	0.537	0.672	0.710	0.581	0.245	0.295	0.345	0.222	0.136	0.182	0.092	0.088
KATE	0.628	0.808	0.762	0.651	0.325	0.378	0.342	0.267	0.174	0.267	0.095	0.114
ProdLDA	0.637	0.780	0.764	0.631	0.374	0.460	0.423	0.289	0.162	0.324	0.080	0.095
RTM	0.543	0.637	0.663	0.574	0.082	0.094	0.126	0.127	0.117	0.194	0.072	0.075
PLANE	0.690	0.799	0.750	0.648	0.417	0.406	0.439	0.288	0.284	0.226	0.107	0.160
NRTM	0.591	0.816	0.549	0.503	0.313	0.404	0.137	0.190	0.149	0.221	0.036	0.049
VGAE	0.671	0.827	0.807	0.718	0.335	0.362	0.495	0.308	0.285	0.265	0.132	0.171

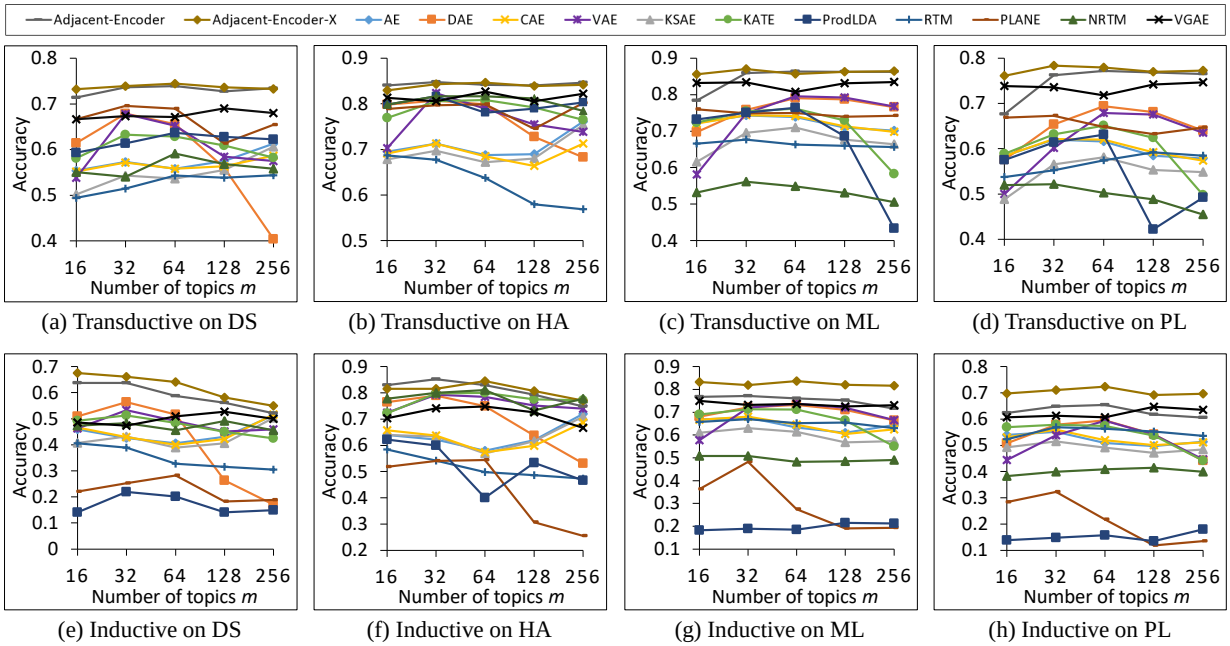
Table 3.3: Transductive results on document classification (left), clustering (middle), and link prediction (right) at $K = 64$.

3.4.2 Transductive Learning

For validating the derived document representations, we rely on three evaluative tasks. The first two are document classification and clustering, evaluated via class labels (these are never part of any learning). The last is link prediction.

Document Classification. Intuitively, topic representations may align with categorizations of documents, i.e., documents within a class may share similar topics. Since our goal is high-quality topic representations, we use simple k-Nearest Neighbors (kNN) as the classifier at the testing phase. For each document, we hide its actual label, and predict its label as the majority label of its k-nearest neighbors based on the Euclidean distance in the low-dimensional topic space. Classification accuracy is used as the metric. The accuracies at $K = 64$ and $10NN$ are summarized in Table 3.3 (left).

Indeed, our models outperform the baselines significantly across all four datasets. Except for ML dataset, *Adjacent-Encoder-X* generally achieves higher results than *Adjacent-Encoder*, because the former captures higher-order proximity in having common neighbors in addition to being direct neighbors. Among the baselines, DAE, VAE, KATE, and VGAE tend to be better,



(a) Transductive on DS (b) Transductive on HA (c) Transductive on ML (d) Transductive on PL

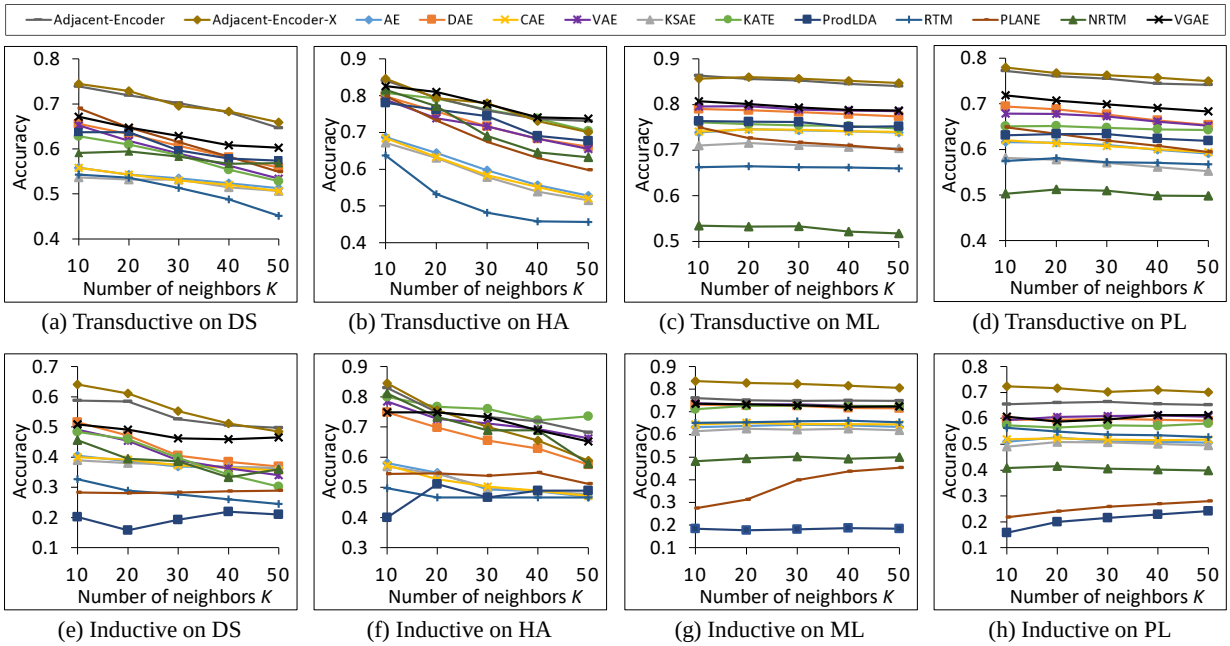
(e) Inductive on DS (f) Inductive on HA (g) Inductive on ML (h) Inductive on PL

Figure 3.3: Transductive and inductive classification accuracy at $10NN$ when varying the number of topics K.

but none achieves a consistent outperformance over others. The top panel of Fig. 3.3 and 3.4 presents the results when varying the number of topics K and neighbors k, respectively. Our models still outperform baseline models most of the time. The only exception is HA, on which our models are competitive with KATE and VGAE. However, the best result of our models as well as baselines is achieved at $10NN$ where both our models outperform all the baselines.

Document Clustering. We can also use the representations for clustering documents, investigating if documents in a cluster tend to share the same class. Class labels are used only for investigating normalized mutual information (NMI) for evaluation. The clustering result at $K = 64$ is shown in Table 3.3 (middle columns). Overall, our models outperform all the baselines significantly. Except for DS, *Adjacent-Encoder-X* achieves better clustering than *Adjacent-Encoder*. Among the baselines, ProdLDA, PLANE, and VGAE tend to perform better than others.

Link Prediction. Given two documents, we could use their topics to predict the link between them. Following [42], the link generation probability is given by $P(\varepsilon_{ij} = 1|\mathbf{z}_i, \mathbf{z}_j) \propto \exp(-||\mathbf{z}_i -$



(a) Transductive on DS (b) Transductive on HA (c) Transductive on ML (d) Transductive on PL

(e) Inductive on DS (f) Inductive on HA (g) Inductive on ML (h) Inductive on PL

Figure 3.4: Transductive and inductive classification accuracy at $m = 64$ when varying the number neighbors K.

$\mathbf{z}_j||^2$). One measure is to examine whether models provide a high generation probability to actual links. We use Mean Average Precision (MAP) as evaluation metric. Following [42], we randomly hide one link for those documents with at least three neighbors (excluding itself) and keep the remaining network connected. The remaining network is present for training. After convergence, we use the topics to predict the held-out links.

Table 3.3 (rightmost) presents the results for $K = 64$ topics. *Adjacent-Encoder* and *Adjacent-Encoder-X* outperform the baselines significantly across four datasets. By comparing our models with AE-based models, we see that considering network structure helps to embed neighbors more closely, thereby achieving a high MAP. Our models rank links higher than others that factor in the network structure (RTM, PLANE, NRTM, VGAE), supporting our outperformance on jointly learning content and network structure.

	Inductive Learning											
Model	Document Classification				Document Clustering				Link Prediction			
	DS	HA	ML	PL	DS	HA	ML	PL	DS	HA	ML	PL
Adjacent-Encoder	0.588	0.830	0.761	0.654	**0.417**	**0.551**	0.477	0.328	**0.421**	**0.462**	0.285	0.218
Adjacent-Encoder-X	**0.640**	**0.845**	**0.836**	**0.724**	0.416	0.489	**0.522**	**0.363**	0.400	0.427	**0.363**	**0.322**
AE	0.405	0.580	0.632	0.509	0.213	0.337	0.340	0.248	0.185	0.233	0.181	0.129
DAE	0.516	0.749	0.732	0.595	0.375	0.436	0.415	0.299	0.347	0.286	0.259	0.198
CAE	0.400	0.573	0.644	0.519	0.212	0.279	0.362	0.253	0.192	0.232	0.185	0.132
VAE	0.491	0.785	0.738	0.594	0.373	0.361	0.404	0.300	0.391	0.346	0.243	0.192
KSAE	0.390	0.569	0.614	0.491	0.269	0.319	0.334	0.232	0.188	0.238	0.148	0.111
KATE	0.484	0.800	0.712	0.573	0.321	0.440	0.354	0.290	0.277	0.336	0.205	0.178
ProdLDA	0.202	0.401	0.184	0.158	0.302	0.292	0.399	0.306	0.220	0.297	0.192	0.140
RTM	0.327	0.498	0.652	0.564	0.000	0.046	0.091	0.048	0.260	0.276	0.210	0.149
PLANE	0.282	0.544	0.275	0.218	0.162	0.192	0.000	0.000	0.306	0.345	0.176	0.134
NRTM	0.456	0.811	0.482	0.408	0.339	0.398	0.167	0.207	0.076	0.097	0.020	0.049
VGAE	0.509	0.748	0.736	0.607	0.280	0.185	0.442	0.291	0.315	0.309	0.237	0.274

Table 3.4: Inductive results on document classification (left), clustering (middle), and link prediction (right) at $K = 64$.

3.4.3 Inductive Learning

For the inductive setting, we evaluate model performance for out-of-sample documents. Thus, we take care not to involve the testing documents during training our encoder models. For training, we observe links only within the training set. For testing, we observe links connecting one testing and one training document, but not links with two testing documents. Once the models are trained, we use their parameters to derive the document representations for test documents and apply the same three evaluative tasks as before.

Table 3.4 shows the results for the inductive setting. The effects of number of topics and neighbors on inductive document classification is shown by Fig. 3.3 and 3.4 (bottom panel). Evidently, similar conclusion as with transductive learning can be drawn that our models are consistently better than baselines. For classification and clustering, PLANE presents satisfying results on transductive, but deteriorates on inductive learning. Among baselines, DAE, VAE, and VGAE tend to outperform others. For link prediction, *Adjacent-Encoder* ranks links higher on DS and HA, while *Adjacent-Encoder-X* performs better on ML and PL.

Model	PMI			
	DS	HA	ML	PL
Adjacent-Encoder	**2.360**	**2.054**	2.180	**2.499**
Adjacent-Encoder-X	1.872	1.887	**2.337**	2.321
AE	0.294	0.446	0.665	0.969
DAE	1.170	1.125	1.203	1.553
CAE	0.348	0.558	0.526	0.684
VAE	0.685	0.793	1.831	1.132
KSAE	0.547	0.285	0.770	0.759
KATE	1.312	1.755	1.619	2.003
ProdLDA	1.638	1.315	1.837	2.088
RTM	1.279	1.678	1.199	1.615
PLANE	1.585	1.847	1.756	2.099
NRTM	1.533	2.041	1.328	1.632

Table 3.5: Topic Coherence PMI when $K = 64$.

Topic	Adjacent-Encoder
1	maze, markov, mdp, observable, minute, intractable, severe, markovian, pomdp, analog
2	move, 0-1, image, nearest, promoter, neighbor, grid, k-nearest, analogy, sketch
3	reward, influence, recurrent, credit, exploratory, max, reinforcement, net, reactive, policy
4	inference, hmm, graphical, practitioner, translation, defense, methodological, causal, probable, assist
5	pair, net, coordination, backpropagation, stronger, broad, network, classic, pendulum, multiclass
Topic	Adjacent-Encoder-X
1	mdp, policy, clarify, identical, pomdp, observable, tradeoff, consequence, noisy, larger
2	cart, selective, exploratory, estimator, phoneme, categorization, stability, multiclass, terminate, axis-parallel
3	parent, graph, substructure, overlap, graphical, load, emulate, integration, fashion, generalisation
4	ica, toronto, detector, blind, maximization, derivation, facial, pca, nonparametric, expansion
5	sigmoidal, shift, logistic, treatment, loop, testing, net, quantify, razor, adversarial

Table 3.6: Top 10 words of 5 randomly selected topics.

3.4.4 Topic Analysis

For better understanding of topic-word association learned by a topic model, we conduct experiments on topic analysis. VGAE is not a topic model, and is not included in this analysis.

Topic Coherence. Our topic-word association is given by $\mathbf{W}$ for *Adjacent-Encoder*, and $\mathbf{W}_1$ for *Adjacent-Encoder-X*. We use PMI [7], defined as $PMI(w_i, w_j) = \log \frac{p(w_i,w_j)}{p(w_i)p(w_j)}$, to evaluate the coherence of predicted words. Using *Google Web 1T 5-gram Version 1* [22], $p(w_i)$ is evaluated from 1-gram corpus, and $p(w_i, w_j)$ from 5-gram corpus. For each topic, we average the

Query Word	Adjacent-Encoder
solution	call, application, problem, solve, provide
neighbor	nearest, k-nearest, paper, describe, artificial
modeling	general, framework, provide, model, knowledge
supervised	learning, task, computational, include, general
speech	recognition, introduce, call, include, paper

Query Word	Adjacent-Encoder-X
dna	protein, produce, attach, promoter, examination
production	unpredictable, manufacture, inventory, discrete-event, key
binary	bit, general, term, include, paper
encode	intelligent, describe, version, representation, form
easily	associate, problem, solve, consider, provide

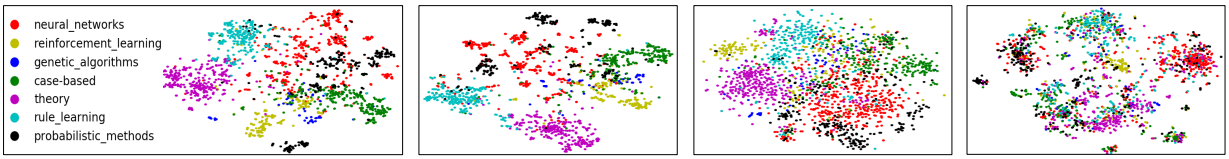
Table 3.7: Top 5 words of 5 randomly selected query words.

pairwise PMI of its top 10 words. For each model, we average PMI of its topics.

Table 3.5 shows that network-based models tend to perform well, benefitting from document relatedness in addition to text content. *Adjacent-Encoder* has higher topic coherence than *Adjacent-Encoder-X* except for ML, presumably in modeling the adjacency vector explicitly the latter may reduce the reconstruction precision of text content. Nevertheless, our models still outperform baselines in most cases.

Topic Interpretability. To gain a semantic sense of topics, we qualitatively present top 10 words of 5 randomly selected topics (Table 3.6). *Adjacent-Encoder*'s topic 2 seems to discuss k-nearest neighbor. Topic 3 discusses reinforcement learning. For *Adjacent-Encoder-X*, topic 1 captures Markov decision problem, while topic 2 seems decision trees.

Each column of topic-word matrix corresponds to a word representation over topics. Thus we check whether similar words are embedded closely. Table 3.7 presents 5 nearest neighbors for each of 5 query words in the word representation space. Both of our models can find relevant words. For example, *Adjacent-Encoder* provides "nearest" and "k-nearest" for the query word *neighbor*, *Adjacent-Encoder-X* presents "protein" and "promoter" for the query word *dna*.



(a) Adjacent-Encoder (b) Adjacent-Encoder-X (c) KATE (d) NRTM

Figure 3.5: t-SNE visualization on ML dataset. (best seen in color)

3.4.5 Visualization

As exploratory analysis, visualization provides an intuitive sense of how topic models embed documents. One may expect a good model to embed documents of a category closely. We apply t-SNE [82] to project 64-dimensional topic space into 2-dimensional visualization space. As a sampler, Fig. 3.5 shows four of the methods on ML dataset. *Adjacent-Encoder* and *Adjacent-Encoder-X* produce good separation between categories.

3.4.6 Extensions and Variants

We investigate the complementarity and potential extensibility of our models by combining proposed architecture with other concepts previously used to enhance AE. For denoising variant, we add Gaussian noise of 0.25 std.dev. to input documents. For our contractive variant, we add Frobenius norm of Jacobian matrix to loss function of our models. For K-sparse variant, as in KSAE and KATE, we keep the values of $\frac{k}{2}$ top positive and $\frac{k}{2}$ top negative hidden neurons and zero others after neighbor competition and before reconstruction.

We test these variants for document classification on ML dataset. We set $K = 64$ and $10NN$. Table 3.8 shows some enhancements tend to produce positive outcomes. Further adding denoising to original models shows the value of denoising. The regularization on loss function by the contractive enhancement provides better results. Indeed *K-Sparse Adjacent-Encoder(-X)* learn competitive representations with original models in terms of classification.

Model	Transductive	Inductive
Adjacent-Encoder	**0.864**	0.761
Denoising Adjacent-Encoder	0.855	**0.780**
Contractive Adjacent-Encoder	0.856	**0.780**
K-Sparse Adjacent-Encoder	0.847	0.765
Adjacent-Encoder-X	0.857	0.836
Denoising Adjacent-Encoder-X	0.865	0.839
Contractive Adjacent-Encoder-X	**0.872**	**0.844**
K-Sparse Adjacent-Encoder-X	0.866	0.823

Table 3.8: Classification accuracy of model variants on ML dataset when $K = 64$ and $10NN$.

3.5 Discussion

We propose *Adjacent-Encoder* and *Adjacent-Encoder-X*, neural topic models that learn unified representations for networked documents. *Adjacent-Encoder* incorporates the network structure implicitly, with similar number of parameters as Auto-Encoder family, yet outperforms the latter. *Adjacent-Encoder-X* that models the network structure explicitly performs even better. Empirical analysis on public datasets support these findings, showcasing the effectiveness of factoring network structure for neural topic modeling. The model extensions, such as denoising, contractive, and sparsity, further improve the performance. We identify two limitations. First, *Adjacent-Encoder-X* has difficulty dealing with large-scale networks, since the dimension of their adjacent vectors becomes quite high, leading to a huge number of parameters. One possible solution is to replace adjacent vectors with low-dimensional vertex embeddings learned by DeepWalk [69]. Second, the proposed models are unsupervised and separate from downstream tasks, e.g., document classification. We need an external classifier to categorize these documents. This two-stage process may influence classification accuracy, since our models and the external classifier have different optimization objectives. To build a model that jointly learns topic representations for documents and also classifies them, we could add a multi-layer perceptron [4] as classifier and jointly optimize both topic modeling and classification losses. We solve this problem in Chapter 5 where our models have both supervised and unsupervised version.

Chapter 4

Dynamic Topic Modeling for Temporal Document Networks

4.1 Introduction

A document network does not emerge suddenly in its entirety. Rather, it is an accumulation of documents created over time. The latent themes in a corpus may also evolve over time, e.g., academic papers track the development of research across years, news articles track the chronology of events. Early attempts to capture document dynamics (DTM [5]) ignore the network aspect.

We postulate that the temporal nature relates not only to when a document is created, but also to how documents created at different times may form linkages. Fig. 4.1 illustrates the formation of a *temporal document network*. Initially at time t_{i-1}, the network contains four documents (A, B, C, and D) and links among them. At time t_i, two new documents, E and F, are published, and bring links to documents B and D, respectively. We use red documents and links to denote the newly appearing data. Moving to time t_{i+1}, document G is published and connected to documents D and F. As time goes by, we observe the growth in terms of both corpus size and network connectivity. Modeling time could better preserve text semantics and network topology. Moreover, time also reveals the possibility of connection. Indicatively,

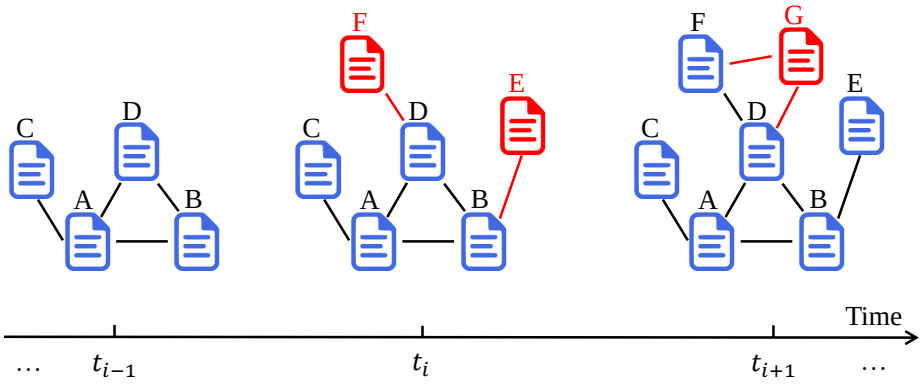


Figure 4.1: Illustration of a temporal document network.

newly published documents are likely to connect to recent neighbors rather than previous ones. Existing topic models for document networks, e.g., RTM [10], focus on the static network. As a result, it may predict a past link using documents published in future time.

Our strategy to better model topics in temporal document networks is a confluence of three factors. *First*, most dynamic topic models [5] deal with the plain text within documents and ignore the network connectivity across documents. However, links constitute additional information on documents' similarities, and modeling them could reveal insightful semantics. *Second*, models for networked documents [10] mainly focus on static networks without considering time. Dynamic process showcases topic evolution and network generation over the time. By modeling it, we may better preserve text semantics and network topology. *Third*, most topic models [2, 10] preserve network structure by modeling first-order neighborhood only, a limited use of network adjacency. The generation of a link between two documents may be influenced by their common historical neighbors, which is higher-order proximity.

Definition 4.1.1 (Temporal Document Network)**.** *Let a temporal document network $\mathcal{G}$ be a tuple $\{\mathcal{D}, \mathcal{E}, \mathcal{T}\}$. $\mathcal{D} = \{\boldsymbol{d}_i\}_{i=1}^N$ contains documents. Each document $\boldsymbol{d}_i \in \mathbb{R}^{|\mathcal{V}|}$ is a vector in the vocabulary space $\mathcal{V}$. $\mathcal{E} = \{\mathcal{E}_t\}_{t=1}^T$ is a set of adjacency matrices. $\mathcal{E}_t \in \mathbb{R}^{N\times N}$ is the adjacency matrix at timestamp t, where $e_{ijt} = 1$ if there is a link between document i and j at timestamp t, $e_{ijt} = 0$ otherwise. T is the maximum timestamp. In this chapter, we consider an undirected network, $e_{ijt} = e_{jit}$. For a document i, its cumulative neighbors observed at timestamp t are*

those directly linked to i from the initial timestamp to t, denoted as $\mathcal{N}_t(i)$. We consider i as its own neighbor, $i \in \mathcal{N}_t(i)$. $\mathcal{T} = \{t_i\}_{i=1}^N$ contains timestamps, t_i is the publication time of document i. If i and j are published at the same time, $t_i = t_j$.

Given $\mathcal{G}$ as input, we propose a neural topic model and derive topic distributions that preserve document semantics $\mathcal{D}$, evolved network structure $\mathcal{E}$, and dynamics $\mathcal{T}$.

4.2 Background

In this work, we use Optimal Transport (OT) for semantic modeling, and Temporal Point Process for dynamic modeling. We first introduce Optimal Transport and Temporal Point Process as preliminary background, after which we discuss the technical details of our proposed models.

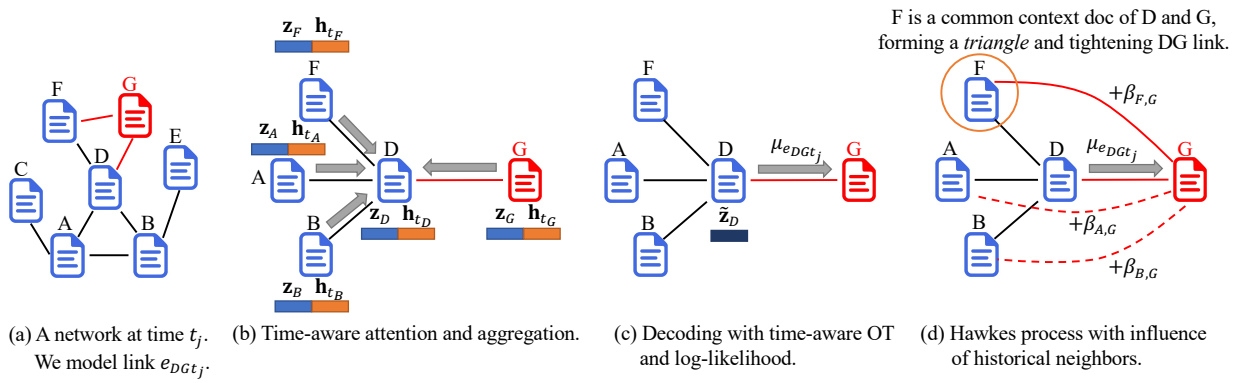
Definition 4.2.1 (Optimal Transport)**.** *Optimal transport measures the distance between two probabilities. Given $\boldsymbol{r} \in \mathbb{R}^{D_r}$ and $\boldsymbol{q} \in \mathbb{R}^{D_q}$, where their respective dimension D_r and D_q may not be the same, their OT distance is*

$$d_{\boldsymbol{C}}(\boldsymbol{r}, \boldsymbol{q}) = \min_{\boldsymbol{P} \in U(\boldsymbol{r}, \boldsymbol{q})} < \boldsymbol{P}, \boldsymbol{C} >= \min_{\boldsymbol{P} \in U(\boldsymbol{r}, \boldsymbol{q})} \sum_{u=1}^{D_r} \sum_{v=1}^{D_q} p_{uv} c_{uv}. \tag{4.1}$$

$\boldsymbol{C} \in \mathbb{R}_{\geq 0}^{D_r \times D_q}$ is cost matrix, each element c_{uv} measures the cost of transport between r_u and q_v. $\boldsymbol{P} \in \mathbb{R}_{>0}^{D_r \times D_q}$ is transport plan. $U(\boldsymbol{r}, \boldsymbol{q}) = \{\boldsymbol{P} \in \mathbb{R}_{>0}^{D_r \times D_q} | \boldsymbol{P}\boldsymbol{1}_{D_q} = \boldsymbol{r}, \boldsymbol{P}^\top \boldsymbol{1}_{D_r} = \boldsymbol{q}\}$ is the transport polytope with $\boldsymbol{r}$ and $\boldsymbol{q}$ as marginals. $\boldsymbol{1}_D$ is a D-dimensional vector with ones. Thus, each element $p_{uv} \in \boldsymbol{P}$ is the probability of transport between r_u and q_v. Given a cost matrix $\boldsymbol{C}$, OT distance between $\boldsymbol{r}$ and $\boldsymbol{q}$ is to find the optimal plan $\boldsymbol{P}^*$ and obtain $d_{\boldsymbol{C}}(\boldsymbol{r}, \boldsymbol{q}) =< \boldsymbol{P}^*, \boldsymbol{C} >$.

We will extend OT to incorporate time and use it to measure the semantic distance between two differently timestamped documents i and j as the probability of the link e_{ijt}.

Definition 4.2.2 (Temporal Point Process)**.** *Temporal point process models the discrete sequential events. It measures the conditional probability $\lambda_\epsilon(t)\Delta t$ of an event ϵ happening in a tiny window $[t, t + \Delta t)$ by assuming that historical events before timestamp t can influence the occur-*



(a) A network at time t_j. We model link e_{DGt_j}.

(b) Time-aware attention and aggregation.

(c) Decoding with time-aware OT and log-likelihood.

(d) Hawkes process with influence of historical neighbors.

Figure 4.2: Illustration of modeling process.

rence of the current event. Here, $\lambda_\epsilon(t)$ is conditional intensity function of event ϵ at timestamp t. Hawkes process is a typical temporal point process. Given historical events $\{\epsilon_h | t_h < t\}$ before timestamp t, its conditional intensity function models the arrival rate of the current event ϵ at timestamp t.

$$\lambda_\epsilon(t) = \mu_\epsilon(t) + \sum_{\epsilon_h : t_h < t} \beta_{\epsilon_h, \epsilon} \kappa(t - t_h), \tag{4.2}$$

$\mu_\epsilon(t)$ is base intensity (the spontaneous arrival rate of the current event ϵ at timestamp t). $\beta_{\epsilon_h,\epsilon}$ is the influence of the historical event ϵ_h on the current ϵ. $\kappa(t - t_h)$ is time decay.

We will use Hawkes process to capture the influence of historical neighbors on the current link formation e_{ijt}.

4.3 Model Architecture and Analysis

4.3.1 NetDTM for Semantic-Level Modeling

We first present the details of NetDTM for semantic-level modeling, and defer the modeling of NetDTM++ to the next subsection.

Each document $\mathbf{d} \in \mathbb{R}^{|\mathcal{V}|}$ is a distribution in the word space, where each element $d_w = \frac{n_w}{\sum_{w' \in \mathcal{V}} n_{w'}}$ is normalized by the length of the document. Here, n_w is the word count of w in the

document. Given document i with its content $\mathbf{d}_i$ on the network, as in [8], we encode it into a K-dimensional topic distribution by $\mathbf{z}_i = \theta(\mathbf{d}_i)$, i.e.,

$$\begin{aligned} \mathbf{z}_i &:= dropout(ReLU(\mathbf{W}_1\mathbf{d}_i + \mathbf{b}_1)), \\ \mathbf{z}_i &:= softmax(batch\_norm(\mathbf{W}_2\mathbf{z}_i + \mathbf{b}_2)). \end{aligned} \tag{4.3}$$

$\mathbf{W}_1 \in \mathbb{R}^{200\times|\mathcal{V}|}$, $\mathbf{W}_2 \in \mathbb{R}^{K\times 200}$, $\mathbf{b}_1 \in \mathbb{R}^{200}$, $\mathbf{b}_2 \in \mathbb{R}^K$ are parameters. We follow [8] to choose 200 as intermediate dimension. $ReLU(x) = \max(0, x)$ and $softmax(x) = \frac{\exp(x_k)}{\sum_{k'=1}^{K}\exp(x_{k'})}$ are activation functions of encoder.

Time-Aware Attention. We seek an attention mechanism to evaluate the importance weights of neighbors. On the one hand, neighbors presenting similar semantics should be assigned high attention values. On the other hand, as mentioned in Section 4.1, since the content of corpus evolves over the time, two linked documents with close publication timestamps are more likely to share similar topics, and should preserve higher attention values. Taking both information into account, we design a time-aware attention mechanism with both semantic similarity and timestamp difference.

$$\tilde{a}_{ij} = tanh([\mathbf{W}_{att}(\mathbf{z}_i||\mathbf{h}_{t_i})]^\top[\mathbf{W}_{att}(\mathbf{z}_j||\mathbf{h}_{t_j})]), \tag{4.4}$$

$$a_{ij} = \frac{\exp(\tilde{a}_{ij})}{\sum_{j'\in\mathcal{N}_t(i)}\exp(\tilde{a}_{ij'})}. \tag{4.5}$$

$(\cdot||\cdot)$ is the concatenation operation, and $\mathbf{W}_{att} \in \mathbb{R}^{K\times 3K}$ is parameter. Attention values are jointly determined by two variables, topic distribution $\mathbf{z}$ at Eq. 4.3 and time embedding $\mathbf{h}_t$ to be discussed shortly. Thus, two documents i and j present a high attention value if their topics are similar, and their publication timestamps are close.

We now define time embedding. Usually, the relative difference between two timestamps, rather than the absolute value of any timestamp, reveals attention values, since the relative timespan informs how close two documents are. Furthermore, the attention at Eq. 4.4 involves the

product of two time embeddings of t_i and t_j. A desirable time embedding should capture timestamp difference when taking product. Inspired by [16], we define

$$\mathbf{h}_t = \sqrt{\frac{1}{K}}[\cos(\omega_1 t), \sin(\omega_1 t), ..., \cos(\omega_K t), \sin(\omega_K t)]^\top. \tag{4.6}$$

Here, $\{\omega_k\}_{k=1}^K$ are parameters. The reason behind such design is

$$\begin{aligned}
\mathbf{h}_{t_i}^\top \mathbf{h}_{t_j} &= \frac{1}{K}[\cos(\omega_1 t_i)\cos(\omega_1 t_j) + \sin(\omega_1 t_i)\sin(\omega_1 t_j) + ... \\
&\quad + \cos(\omega_K t_i)\cos(\omega_K t_j) + \sin(\omega_K t_i)\sin(\omega_K t_j)] \\
&= \frac{1}{K}[\cos(\omega_1(t_i - t_j)) + ... + \cos(\omega_K(t_i - t_j))] \\
&\approx \mathbb{E}_\omega[\cos(\omega(t_i - t_j))].
\end{aligned} \tag{4.7}$$

Thus, the product of two time embeddings is transformed into the timestamp difference, which aligns with our requirement.

Linked documents tend to share similar topics, e.g., cited papers discuss similar research problems. We aggregate topics of document i's neighbors to itself and obtain

$$\tilde{\mathbf{z}}_i = \sum_{j \in \mathcal{N}_t(i)} a_{ij} \mathbf{z}_j. \tag{4.8}$$

At Fig. 4.2(a), document G is published at time t_j. We model link e_{DGt_j} as an illustration. Above time-aware attention is shown by Fig. 4.2(b) where we aggregate topics of neighbors to document D.

Time-Aware Optimal Transport. We base the semantic modeling on Optimal Transport (OT), because it has achieved promising results in neural topic modeling [118]. Here, we are motivated to incorporate time into OT. A document i is usually represented by two distributions, latent topic distribution $\tilde{\mathbf{z}}_i$ and observed content $\mathbf{d}_i$. They should consistently reflect the same document. We thus seek to minimize the OT semantic distance between latent topic distribution

$\tilde{\mathbf{z}}_i$ and word distribution $\mathbf{d}_i$, i.e., $\min d_{\mathbf{C}}(\tilde{\mathbf{z}}_i, \mathbf{d}_i)$.

Since we are interested in network modeling, and links indicate a similar latent semantics of two documents, we now allow OT to push topic distribution $\tilde{\mathbf{z}}_i$ also to document i's neighbors $j \in \mathcal{N}_t(i)$, i.e., $\min d_{\mathbf{C}}(\tilde{\mathbf{z}}_i, \mathbf{d}_j)$. Since documents are sequentially organized, and the recently published documents link to previous ones, the publication timestamps t_i and t_j may not be the same. However, original optimal transport at Eq. 4.1 does not preserve such time information. To model document dynamics, we now propose Time-Aware Optimal Transport, which also takes timestamps as inputs.

$$\min d_{\mathbf{C}}(\tilde{\mathbf{z}}_i, \mathbf{d}_j, t_i, t_j), \tag{4.9}$$

where $j \in \mathcal{N}_t(i)$ is document i's neighbors at timestamp t.

Definition 4.3.1 (Time-Aware Optimal Transport)**.** *Given $\tilde{\boldsymbol{z}}_i \in \mathbb{R}^K$ and $\boldsymbol{d}_j \in \mathbb{R}^{|\mathcal{V}|}$, $\tilde{\boldsymbol{z}}_i$ with timestamp t_i, and $\boldsymbol{d}_j$ with timestamp t_j (without loss of generality, $t_j \geq t_i$), the time-aware OT distance is*

$$d_{\boldsymbol{C}}(\tilde{\boldsymbol{z}}_i, \boldsymbol{d}_j, t_i, t_j) = \min_{\boldsymbol{P} \in U(\tilde{\boldsymbol{z}}_i, \boldsymbol{d}_j, t_i, t_j)} \sum_{t=t_i}^{t_j} \sum_{k=1}^{K} \sum_{w=1}^{|\mathcal{V}|} p_{tkw} c_{tkw}. \tag{4.10}$$

Here, $\boldsymbol{C} \in \mathbb{R}_{\geq 0}^{(t_j - t_i + 1) \times K \times |\mathcal{V}|}$ is cost matrix, and each element c_{tkw} measures the cost of transport between topic k and word w at timestamp t. $\boldsymbol{P} \in \mathbb{R}_{>0}^{(t_j - t_i + 1) \times K \times |\mathcal{V}|}$ is transport plan. $U(\tilde{\boldsymbol{z}}_i, \boldsymbol{d}_j, t_i, t_j) = \{\boldsymbol{P} \in \mathbb{R}_{>0}^{(t_j - t_i + 1) \times K \times |\mathcal{V}|} | \boldsymbol{1}_{(t_j - t_i + 1)} \boldsymbol{P} \boldsymbol{1}_{|\mathcal{V}|} = \tilde{\boldsymbol{z}}_i, (\boldsymbol{1}_{(t_j - t_i + 1)} \boldsymbol{P})^\top \boldsymbol{1}_K = \boldsymbol{d}_j\}$ is the transport polytope with $\tilde{\boldsymbol{z}}_i$ and $\boldsymbol{d}_j$ as marginals. $\boldsymbol{1}_D$ is a D-dimensional vector with ones. Thus, each element $p_{tkw} \in \boldsymbol{P}$ measures the probability of transport between topic k and word w at timestamp t. Given a cost matrix $\boldsymbol{C}$, time-aware OT distance between $\tilde{\boldsymbol{z}}_i$ and $\boldsymbol{d}_j$ is to find the optimal plan $\boldsymbol{P}^*$ and obtain $d_{\boldsymbol{C}}(\tilde{\boldsymbol{z}}_i, \boldsymbol{d}_j, t_i, t_j) = < \boldsymbol{P}^*, \boldsymbol{C} >$.

Comparing Eq. 4.10 with original OT at Eq. 4.1, in addition to the summation over topics and words, we further add one more time dimension for summation across the timespan $t_j - t_i + 1$. Thus, time-aware OT measures the semantic distance between topic distribution $\tilde{\mathbf{z}}_i$ and word distribution $\mathbf{d}_j$ across the timespan. Original OT becomes a special case of time-aware OT when

Algorithm 1 Time-Aware Sinkhorn Iteration

Input: Document i's topic distribution $\tilde{\mathbf{z}}_i$, neighbor j's word distribution $\mathbf{d}_j$ ($j \in \mathcal{N}_t(i)$), timestamp t_i and t_j, cost matrix $\mathbf{C}$, γ.

Output: OT plan $\mathbf{P}^*$, time-aware OT distance $d_{\mathbf{C}}(\tilde{\mathbf{z}}_i, \mathbf{d}_j, t_i, t_j)$.

1: Initialize $\mathbf{\Psi}_1 = \frac{\mathbf{1}_K}{K}$, $\mathbf{t} = \frac{\mathbf{1}_{(t_j - t_i + 1)}}{t_j - t_i + 1}$, and $\mathbf{\Phi} = \exp(-\frac{\mathbf{C}}{\gamma})$.
2: **while** not converged **do**
3: $$\mathbf{\Psi}_2 = \frac{\mathbf{d}_j}{(\mathbf{t}\mathbf{\Phi})^\top \mathbf{\Psi}_1}, \qquad \mathbf{\Psi}_1 = \frac{\tilde{\mathbf{z}}_i}{(\mathbf{t}\mathbf{\Phi})\mathbf{\Psi}_2}.$$
4: **end while**
5: Obtain OT plan $\mathbf{P}^* = \text{diag}(\mathbf{\Psi}_1)(\mathbf{t}\mathbf{\Phi})\text{diag}(\mathbf{\Psi}_2)$
6: Obtain time-aware OT $d_{\mathbf{C}}(\tilde{\mathbf{z}}_i, \mathbf{d}_j, t_i, t_j) = \text{diag}(\mathbf{\Psi}_1)(\mathbf{t}(\mathbf{\Phi} \otimes \mathbf{C}))\text{diag}(\mathbf{\Psi}_2)$. Here $\otimes$ is element-wise product.

document i and j are published at the same timestamp $t_i = t_j$ with no timespan. Intuitively, two documents that are semantically similar and published closely should present a low OT distance, since they do not transport with much semantic cost across a long timespan, and vice versa. Therefore, we are able to use time-aware OT to measure the probability of link e_{ijt}. Lower the distance, higher the probability.

We now define the semantic cost matrix $\mathbf{C}$. Each element c_{tkw} represents the semantic dissimilarity between topic k and word w at timestamp t. It can be cosine dissimilarity or Euclidean distance. In this work, we choose the former and define $c_{tkw} = 1 - \cos(\mathbf{g}_{tk}, \mathbf{e}_w)$, where $\mathbf{g}_{tk}$ is the randomly initialized embedding for topic k at time t, and $\mathbf{e}_w$ is the pre-trained word embedding, such as GloVe [68] or word2vec [61]. Therefore, external knowledge is naturally incorporated into our model and helps the semantic modeling.

Evaluating OT distance requires to obtain the optimal plan $\mathbf{P}^*$, which can be calculated by Sinkhorn iteration [14]. Since we extend the original OT to incorporate time information, we correspondingly propose Time-Aware Sinkhorn Iteration at Algo. 1.

Decoding. Time-aware optimal transport at Eq. 4.9 pushes topic distribution $\tilde{\mathbf{z}}_i$ to neighboring word distribution $\mathbf{d}_j$, which is similar to a generative decoding process. We further explicitly

design a decoder below to generate the textual content of neighbors.

$$\hat{\mathbf{d}}_j = \frac{1}{t_j - t_i + 1} \sum_{t=t_i}^{t_j} softmax((2 - \mathbf{C}_t)^\top \tilde{\mathbf{z}}_i). \tag{4.11}$$

Here, $\mathbf{C}_t \in \mathbb{R}^{K \times |\mathcal{V}|}$ is the t^{th} slice of semantic cost matrix $\mathbf{C}$, which captures topic-word distribution and is used as decoding parameter. We average the output across the timespan $t_j - t_i + 1$ as the generated content. We obtain log-likelihood, $l(\mathbf{d}_j, \hat{\mathbf{d}}_j) = \mathbf{d}_j^\top \log \hat{\mathbf{d}}_j$, of the generative process. Finally, as in [118], combining log-likelihood and time-aware OT, we have the following loss function.

$$\begin{aligned} \mu_{e_{ijt}} &= l(\mathbf{d}_j, \hat{\mathbf{d}}_j) - \eta_{OT} d_{\mathbf{C}}(\tilde{\mathbf{z}}_i, \mathbf{d}_j, t_i, t_j), \\ L_{NetDTM} &= -\sum_{t=1}^{T} \sum_{e_{ijt} \in \mathcal{E}_t} \mu_{e_{ijt}} + \eta_p L_p. \end{aligned} \tag{4.12}$$

Hyperparameter η_{OT} balances log-likelihood and time-aware OT, and $\mu_{e_{ijt}}$ measures the probability of link e_{ijt}. At Fig. 4.2(c), we generate the content of document G with time-aware OT and log-likelihood as semantic modeling of link e_{DGt_j}.

Different timestamps have their own topic embeddings $\{\mathbf{g}_{tk}\}_{t=1}^T$ and topic-word distributions $\{\mathbf{C}_t\}_{t=1}^T$. To associate the modeling process of different timestamps and capture topic evolution across the whole time period, we seek to chronologically *chain* the topics. Following [5, 18], we thus draw topic embeddings using a Markov chain with Gaussian distribution, $\mathbf{g}_{tk} \sim p(\mathbf{g}_{tk}|\mathbf{g}_{t-1,k}) = \mathcal{N}(\mathbf{g}_{t-1,k}, \sigma^2 \mathbf{I})$ for $t = 2, ..., T$ and $k = 1, ..., K$. Its log-likelihood is

$$\log p(\mathbf{g}_{tk}|\mathbf{g}_{t-1,k}) \propto -\frac{1}{2\sigma^2} ||\mathbf{g}_{tk} - \mathbf{g}_{t-1,k}||^2 = \eta_p ||\mathbf{g}_{tk} - \mathbf{g}_{t-1,k}||^2. \tag{4.13}$$

We set $\eta_p = -\frac{1}{2\sigma^2}$. Summing all the timestamps and topics, we obtain $\eta_p L_p = \eta_p \sum_{t=2}^T \sum_{k=1}^K ||\mathbf{g}_{tk} - \mathbf{g}_{t-1,k}||^2$. Adding such a prior term to the loss function at Eq. 4.12 as a regularizer, we obtain multiple topic embeddings $\{\mathbf{g}_{tk}\}_{t=1}^T$, which capture topic evolution.

4.3.2 NetDTM++ for Network-Level Modeling

The above process captures network connectivity by semantically generating textual content of neighbors. Such a generative process measures the semantic similarity between two documents, which is also the *spontaneous* probability of an event (the natural establishment of link e_{ijt}), without considering the impact of historical events. However, in addition to the internal semantics, we discover that the formation of link e_{ijt} is also externally influenced by the existing network topological structure generated so far. For example, two academic papers with a lot of common citations also likely cite each other, and such common citations enhance the possibility of their similar research topics. While NetDTM captures semantic modeling, here we propose an extended model, NetDTM++, to also incorporate the impact of network topology. Thus, we model the impact of previous links on the current link.

Hawkes Process Modeling. For a document i, we apply the same encoding process at Eq. 4.3 to obtain its topic distribution $\mathbf{z}_i = \theta(\mathbf{d}_i)$. After time-aware attention at Eq. 4.4–4.5 and Eq. 4.8, we obtain its aggregated topic distribution $\tilde{\mathbf{z}}_i$. To model the effect of previously generated links $\{e_{t_h}\}_{t_h \leq t}$ on the establishment of the current link e_{ijt}, we design a Hawkes Process, i.e.,

$$\lambda_{e_{ijt}} = \mu_{e_{ijt}} + \eta_{HP} \sum_{e_{t_h}: t_h \leq t} \beta_{e_{t_h}, e_{ijt}} \kappa(t - t_h). \tag{4.14}$$

$\mu_{e_{ijt}}$ is base intensity obtained at Eq. 4.12, containing time-aware OT and log-likelihood. The second term models the impact of previously established links, where $\beta_{e_{t_h}, e_{ijt}}$ is the influence of a previous link e_{t_h} on the current e_{ijt}, and $\kappa(t - t_h)$ is time decay term. Hyperparameter η_{HP} balances the semantic and network modeling. NetDTM becomes a special case of NetDTM++ when $\eta_{HP} = 0$.

However, the second term requires the summation over the entire link set generated so far, which is inefficient in computation. Moreover, usually neighbors of document i influence the formation of e_{ijt} the most; links multiple hops away from e_{ijt} almost have no impact, but likely

bring noisy information. Thus, we modify Eq. 4.14 to only consider links between i and its neighbors, but not all the links. This process models the second-order proximity at Fig. 4.2(d).

$$\lambda_{e_{ijt}} = \underbrace{\mu_{e_{ijt}}}_{\text{semantic modeling}} + \eta_{HP} \underbrace{\sum_{p \in \mathcal{N}_t(i)} \beta_{pj} a_{ip}}_{\text{network modeling}} . \tag{4.15}$$

β_{pj} models the influence of second-order proximity p, which represents the surrounding network context between i and j. As mentioned, two documents sharing similar contextual vertices should preserve a high semantic similarity. At Fig. 4.2(d), document F is a common context of D and G. Such network structure forms a triangle, which tightens the link between D and G and enhances their semantic similarity. Following LINE [77], to model second-order proximity, we introduce a context embedding $\mathbf{w}_p \in \mathbb{R}^K$ and define

$$\beta_{pj} = \log \sigma(\mathbf{w}_p^\top \tilde{\mathbf{z}}_j) + \sum_{m=1}^{M} \mathbb{E}_{d \sim \Pr_n(d)} [\log \sigma(-\mathbf{w}_d^\top \tilde{\mathbf{z}}_j)]. \tag{4.16}$$

$\sigma(x) = \frac{1}{1+\exp(-x)}$ is sigmoid, M is the number of negative samples, $\Pr_n(d)$ is a noise distribution over documents. A high value of β_{pj} increases $\lambda_{e_{ijt}}$, the log-likelihood of the link. At Fig. 4.2(d), in addition to the semantic decoding, document D's neighbors also influence G by adding context $\beta_{\cdot,G}$. Higher-order proximity is modeled.

Time-aware attention a_{ip} at Eq. 4.15 measures the importance of neighbors, including semantic similarity and timestamp difference. Since a_{ip} already contains time difference, we do not design an extra time decay term. Finally, the loss function of NetDTM++ is

$$L_{NetDTM++} = -\sum_{t=1}^{T} \sum_{e_{ijt} \in \mathcal{E}_t} \lambda_{e_{ijt}} + \eta_p L_p. \tag{4.17}$$

We use minibatch gradient descent with Adam [36] optimizer to minimize loss functions. After training convergence, we infer the topic distribution of a previously unseen document $\mathbf{d}'$ by

$\mathbf{z}' = \theta(\mathbf{d}')$.

Complexity Analysis. Encoding is $\mathcal{O}(200(|\mathcal{V}| + K))$. Time-aware attention has $\mathcal{O}(K^2 + \deg_{\max} K)$ where $\deg_{\max}$ is the maximum degree of a document on the network. Time-aware OT is $\mathcal{O}(WK|\mathcal{V}|T)$. W is the dimension of word embeddings. Decoding is $\mathcal{O}(K|\mathcal{V}|T)$. Hawkes process is $\mathcal{O}(K^2 + KN)$. Putting all components together, we have $\mathcal{O}(200(|\mathcal{V}| + K) + K^2 + WK|\mathcal{V}|T)$ for NetDTM, and $\mathcal{O}(200(|\mathcal{V}| + K) + K^2 + WK|\mathcal{V}|T + KN)$ for NetDTM++.

4.4 Experiments

The goal of experiments is to evaluate the topics learned by our models against baseline models by evaluation tasks, such as document classification, link prediction, topic analysis, etc.

Datasets. Cora [58] is a citation network with abstracts as content and citations as links. Each paper has a publication year. Following [122], we create two independent datasets, Machine Learning (**ML**) and Programming Language (**PL**). ML papers are published between 1989 and 1998, and PL between 1987 and 1999. **HEP-TH** [46] is another citation network of Physics papers published from January 1993 to April 2003. Timestamp can be defined by season, half year, or year, resulting in 46, 23, or 12 timestamps, respectively. **Web** [45] is a Web page hyperlink network. Each page contains frequent phrases of a news article between August and December 2008. Timestamp can be defined by semimonthly or monthly, resulting in 10 or 5 timestamps. For the following experiments, we use yearly timestamp for ML, PL, and HEP-TH, since academic conferences are usually held annually. For Web, we use semimonthly as timestamp, due to the transience of news articles. We will investigate the effect of timestamp granularity. Table 4.1 shows the statistics.

Baselines. We compare against four categories of baselines. *i*) **Static topic models without networks**, including ProdLDA [76], WLDA [63], and NSTM [118]. WLDA applies Wesserstein distance for topic modeling, and NSTM is a neural model with optimal transport. These models do not incorporate document dynamics or network connectivity. *ii*) **Dynamic topic models**,

Name	#Documents	#Links	Vocabulary	#Labels	#Timestamps
ML	1,489	3,474	3,302	7	10
PL	1,424	3,955	3,062	9	13
HEP-TH	27,770	352,285	3,027	N.A.	12
Web	188,741	207,963	5,000	N.A.	10

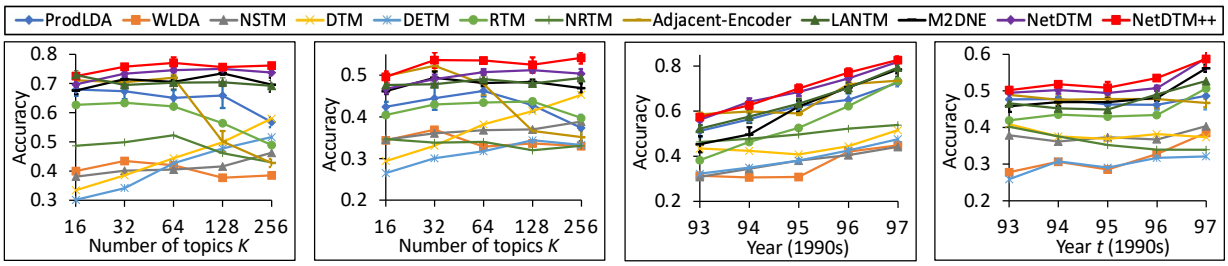
Table 4.1: Dataset statistics.

including DTM [5] and DETM [18]. DTM extends LDA [6] in a dynamic setting. DETM extends VAE [38] and leverages pre-trained word embeddings for dynamic modeling. They indeed incorporate time, but ignore the document adjacency. Thus by comparing to them, we highlight the advantage of jointly modeling dynamics and network structure. *iii*) **Topic models for document networks**, such as graphical model RTM [10], and neural models NRTM [2] and Adjacent-Encoder [108]. They consider textual content and network structure for modeling, but no one models dynamic process of documents. By comparison, we show the effectiveness of dynamic modeling. *iv*) **Temporal graph embedding** learns node embeddings on temporal graphs in an unsupervised way. Strictly speaking, they are not topic models, nor baselines. For completeness, we still compare to M2DNE [54].

Implementation Details. Hyperparameters are set based on the result on validation set (see below document classification on how to split it). We set 2 as Dirichlet prior for RTM. For models with word embeddings (NSTM, DETM, and ours), we use 300D GloVe. For our models, $\eta_{OT} = \eta_{HP} = \eta_p = 1$ after searching in $[0.5, 1, 2, 4, 10]$. We set dropout rate to 0.75, $\gamma = 20$, and $M = 5$. Each result is obtained by 5 independent runs. We report both average and std.dev.

4.4.1 Quantitative Evaluation

Document Classification. Documents from the same category should preserve similar topics. A good topic model should learn similar topics to group such documents and separate different categories. Following LDA [6], we conduct document classification to evaluate topic quality. Since we observe the dynamic process of network evolution, we split the datasets using timestamps.



(a) Number of topics on ML (b) Number of topics on PL (c) Different years (1990s) on ML (d) Different years (1990s) on PL

Figure 4.3: Classification accuracy w.r.t. (a-b) different number of topics, (c-d) different years.

Specifically, we split documents published before timestamp T (inclusive) for training (among which 10% are reserved for validation) where T is the maximum timestamp in the training set. We observe training documents and links among them during training process. Labels are never involved for training. After convergence, we infer topic distributions of test documents published after timestamp T (exclusive). We apply kNN classifier [4] for document classification where we input topic distributions of training documents to train the classifier, and predict the labels of test documents.

We first vary the number of topics K from 16 to 256, and report classification accuracy with 5NN on ML and PL dataset at Fig. 4.3(a-b). Here, we train the models using documents and links generated before year $T = 1996$ (inclusive), and predict the labels of documents after $T = 1996$ (exclusive). Such timestamp provides around 80/20 split. For clarity, we show std.dev. of our models and best baselines only. Overall, our models perform stably across different number of topics. Our models, Adjacent-Encoder, and M2DNE show better results than others, since network connectivity indicates similarities among documents. Ours and M2NDE are generally better than Adjacent-Encoder. We attribute this outperformance to the modeling of dynamic process. Our models show 3-4% improvement over the best baseline, M2DNE. Since most models present an increasing performance before 64 topics, after which some keep flat, while others deteriorate, we keep $K = 64$ for the following experiments.

We then vary the observed timestamps T from 1993 to 1997, and present the accuracy with

5NN and 64 topics at Fig. 4.3(c-d) for ML and PL, respectively. Horizontal axis corresponds to different years T. The goal is to investigate how models perform when we observe different number of timestamps. As time goes by, the network becomes larger and we observe more timestamps for training, thus the accuracy of most models presents an increasing trend. At $T = 1993$ where only a few timestamps are observed, our models show a competitive performance with Adjacent-Encoder, a static document network model, since our models can not make full use of the dynamic information. After moving to recent years, we discover a significant improvement of our models over Adjacent-Encoder, due to the benefit of document dynamics. NetDTM++ generally classifies documents more accurately than NetDTM, due to Hawkes process, which also models the influence of historical events.

Link Prediction. A topic model should well preserve network structure and encode potentially linked documents closely. Following RTM [10], we predict the links on the network. We split the datasets the same as classification. We observe training documents and links within them for training. We infer topics of test documents and predict the links among them. As in [42], the probability of a link is $p(e_{ij}|\mathbf{z}_i, \mathbf{z}_j) \propto \exp(-||\mathbf{z}_i - \mathbf{z}_j||^2)$. We use mean average precision (MAP) [108] as metric to compare the predicted probability and ground-truth connectivity. As in classification, we also vary the number of observed timestamps. Since different datasets have different timespan, some cross many years, while others cross a few months, for consistency purpose across four datasets, we vary the percentage of total timestamps from 40% to 80%. Here, 40% means we observe documents and links generated in previous 40% timestamps, and predict links among documents in the future.

Table 4.2 shows that as the percentage of timestamps increases, most models improve their results, because they observe more documents and links for training. Again, when we observe only 40% timestamps, our models are competitive with Adjacent-Encoder on HEP-TH. When the observed timestamps accumulate to 80%, our models present a significant improvement. This enhances the benefit of dynamics in our models as compared to static network models. Compared

Model	ML			PL		
	40%	60%	80%	40%	60%	80%
ProdLDA	6.5±0.4	10.8±0.3	20.1±1.1	6.9±0.3	11.7±0.8	19.3±1.2
WLDA	3.2±0.5	3.4±0.3	7.0±1.0	2.9±0.2	4.3±0.5	7.0±2.1
NSTM	3.6±0.4	5.7±0.4	10.1±2.2	3.9±0.1	7.2±0.4	12.1±0.7
DTM	6.9±0.3	7.9±0.6	13.8±2.9	7.1±0.2	10.2±0.8	13.5±1.5
DETM	1.7±0.0	9.6±0.4	15.6±2.8	4.5±0.4	6.9±1.5	8.0±1.7
RTM	10.4±0.2	15.0±0.4	24.3±0.7	9.5±0.2	15.1±0.5	21.3±0.8
NRTM	6.2±0.4	7.0±0.6	10.6±1.6	6.2±0.5	8.3±0.6	9.0±0.3
Adjacent-Encoder	10.3±0.4	16.3±0.6	26.3±0.7	7.8±0.5	18.9±0.4	22.2±0.4
M2DNE	3.1±0.0	7.2±0.0	16.4±0.2	3.5±0.0	12.2±0.0	16.1±0.2
NetDTM	**12.2±0.4**	17.3±0.8	25.8±0.7	11.2±0.4	17.8±0.5	24.0±0.4
NetDTM++	12.0±0.3	**18.0±0.2**	**28.3±1.0**	**11.5±0.3**	**19.9±0.3**	**26.8±0.8**

Model	HEP-TH			Web		
	40%	60%	80%	40%	60%	80%
ProdLDA	0.4±0.0	0.5±0.0	1.3±0.2	10.7±0.3	10.7±0.3	10.7±0.3
WLDA	0.5±0.0	0.7±0.1	2.1±0.2	7.0±0.0	9.6±0.1	11.7±0.2
NSTM	0.6±0.0	0.8±0.0	1.7±0.1	1.1±0.1	1.3±0.1	1.7±0.0
DTM	2.1±0.0	3.3±0.0	6.5±0.3	3.7±0.0	4.1±0.0	4.4±0.0
DETM	2.7±0.0	3.2±0.0	5.3±0.2	**13.5±0.0**	14.7±0.0	16.1±0.0
RTM	3.3±0.1	4.1±0.1	7.0±0.2	11.1±0.1	12.4±0.0	13.8±0.0
NRTM	0.6±0.0	0.6±0.0	1.2±0.0	0.5±0.0	0.5±0.0	1.0±0.3
Adjacent-Encoder	**6.1±0.1**	**7.9±0.2**	13.3±0.2	**13.5±0.0**	14.5±0.0	14.8±0.1
M2DNE	4.3±0.0	5.6±0.0	10.1±0.0	0.5±0.0	0.7±0.0	1.0±0.0
NetDTM	4.8±0.1	6.2±0.1	11.4±0.1	**13.5±0.0**	14.9±0.0	**16.7±0.1**
NetDTM++	5.7±0.0	7.3±0.1	**14.0±0.3**	**13.5±0.0**	**15.0±0.0**	**16.7±0.1**

Table 4.2: Link prediction MAP at $K = 64$ (results are in percentage) when varying the percentage of total timestamps.

to models without network structure, we emphasize that incorporating network can bring useful information.

4.4.2 Topic Analysis

Perplexity. Following LDA [6] and DTM [5], we conduct perplexity experiment to evaluate topic quality. For dynamic topic models, i.e., DTM, DETM, and ours, we obtain a series of topic-word distributions $\{2 - \mathbf{C}_t\}_{t=1}^{T}$, which capture topic evolution over the time. The latest

Model	ML			PL		
	40%	60%	80%	40%	60%	80%
ProdLDA	8.16±0.00	8.07±0.00	8.07±0.00	8.15±0.00	8.03±0.00	8.02±0.00
WLDA	8.63±0.10	8.60±0.14	8.12±0.99	8.62±0.16	8.34±0.03	8.49±0.44
NSTM	8.05±0.00	7.99±0.00	7.97±0.00	7.97±0.00	7.90±0.00	7.89±0.00
DTM	8.10±0.00	8.10±0.00	8.10±0.00	8.02±0.00	8.02±0.00	8.02±0.00
DETM	11.66±0.18	11.63±0.25	9.63±0.16	8.78±0.09	8.20±0.07	8.06±0.05
RTM	7.99±0.02	7.97±0.01	7.90±0.02	7.84±0.03	7.72±0.05	**7.65±0.02**
NRTM	33.10±0.44	28.99±0.13	22.72±0.27	38.61±2.32	33.43±0.16	30.19±0.14
Adjacent-Encoder	7.94±0.01	7.99±0.02	8.07±0.03	7.82±0.08	7.76±0.09	7.97±0.04
NetDTM	**7.90±0.02**	7.89±0.05	7.89±0.04	7.78±0.01	7.81±0.03	7.89±0.05
NetDTM++	**7.90±0.02**	**7.80±0.02**	**7.79±0.02**	**7.72±0.01**	**7.69±0.02**	7.72±0.03

Model	HEP-TH			Web		
	40%	60%	80%	40%	60%	80%
ProdLDA	8.58±0.05	8.60±0.00	8.71±0.00	8.52±0.00	8.52±0.00	8.52±0.00
WLDA	44.30±0.24	44.50±0.33	42.98±0.07	44.43±0.02	43.73±0.04	44.09±0.06
NSTM	7.93±0.00	7.93±0.00	7.93±0.00	8.28±0.00	8.27±0.00	8.28±0.00
DTM	8.01±0.00	8.01±0.00	8.01±0.00	11.61±0.07	11.59±0.07	11.61±0.10
DETM	8.28±0.09	8.09±0.05	7.91±0.06	10.39±0.19	9.46±0.05	8.84±0.14
RTM	7.81±0.00	7.81±0.00	7.81±0.00	20.68±1.51	18.73±0.95	9.87±0.12
NRTM	22.17±0.12	21.32±0.13	21.21±0.34	33.56±0.76	33.56±0.76	38.43±1.33
Adjacent-Encoder	7.86±0.03	7.86±0.03	7.89±0.07	8.72±0.09	8.72±0.09	8.72±0.09
NetDTM	7.79±0.06	7.81±0.06	7.96±0.19	8.79±0.22	8.79±0.06	8.69±0.05
NetDTM++	**7.77±0.02**	**7.79±0.02**	**7.77±0.01**	**8.13±0.11**	**7.92±0.08**	**8.11±0.21**

Table 4.3: Perplexity experiment at $K = 64$ when varying the percentage of total timestamps. Lower is better.

distribution $2 - \mathbf{C}_T$ can best represent the current topic-word distribution. To generalize to future documents published after timestamp T, we should use $2 - \mathbf{C}_T$. This is consistent with [5]. Because perplexity, $\exp\{-\frac{\log \Pr(D_{test})}{\sum_{d' \in D_{test}} N_{d'}}\}$, is exponential and varies w.r.t. its power, we show the power, $-\frac{\log \Pr(D_{test})}{\sum_{d' \in D_{test}} N_{d'}}$. Lower is better. Again, we vary the percentage of timestamps from 40% to 80% and show the results at Table 4.3. M2DNE is not a topic model, thus cannot evaluate perplexity. Network models (RTM and Adjacent-Encoder) perform the best among baselines. Network structure can capture the case where two documents are different in observed content, but consistent in latent semantics, thus incorporating network can help semantic learning and improve topic modeling. Due to dynamic modeling, DTM provides decent results. By combining both network effect and dynamics, our models outperform baselines significantly. NetDTM++ is

generally better than NetDTM, since Hawkes process with the influence of historical events can better encode semantically similar document closely.

Topic Coherence. Decoding parameter $2 - \mathbf{C}_t \in \mathbb{R}^{K \times |V|}$ captures the keywords of each topic. Each row is the distribution of a topic over the vocabulary. The keywords of that topic are those with the highest values on that row. Following ProdLDA [76], we evaluate the coherence of top-10 keywords of each topic and report NPMI. We use *Google Web 1T 5-gram Version 1* [22] as external corpus. Table 4.4 shows the results. M2DNE is not a topic model and is excluded. Benefiting from optimal transport, NSTM is the best model among baselines. By comparing to it, our models extend OT to incorporate dynamic information, and improve the performance.

Topic Evolution. To intuitively understand how our models capture topic evolution, Fig. 4.4 shows the plot of our models on ML dataset. Horizontal axis represents different years, and vertical axis is the word probability in $\{2 - \mathbf{C}_t\}_{t=1989}^{1998}$. Four lines represent randomly selected keywords of the same topic. For NetDTM, "algorithm and compression" remained a popular research over the years. But researchers gradually shifted their focus away from "text indexing", potentially because topic models (PLSA [30]) was proposed, and traditional indexing method became inefficient and less attractive. For NetDTM++, "probabilistic bayesian inference" attracted much attention over the years, while "regression model" fluctuated and gradually decayed, which is possibly because neural network started to present its ability as a universal approximator, and traditional regression models became less interesting.

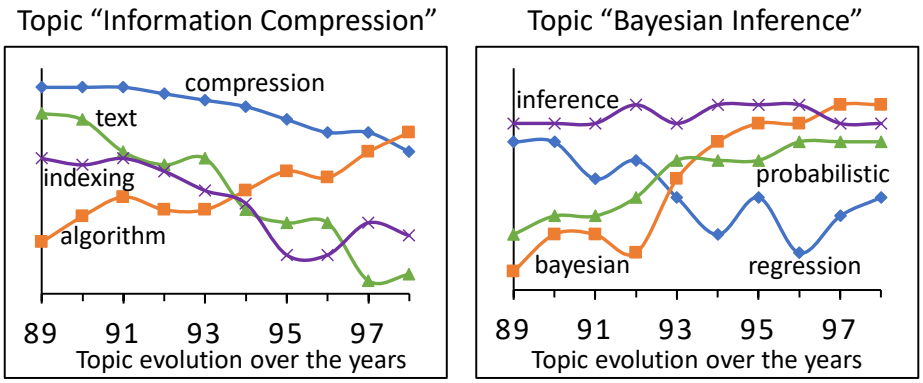
4.4.3 Model Analysis

To better understand our models, we conduct ablation analysis here.

Effect of Network Structure. We randomly remove a proportion of links on the network. We vary the percentage of remaining observed links and report the classification accuracy on ML dataset at Fig. 4.5(a). As we observe more links, the performance tends to increase. Compared to the case with no observed links, adding a small proportion of links can significantly boost the

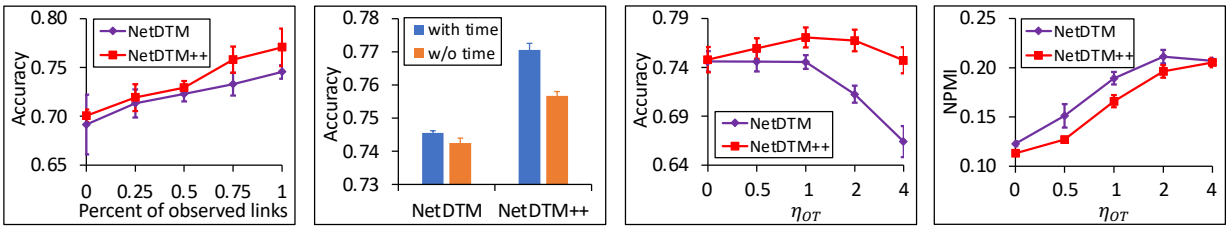
Model	ML	PL	HEP-TH	Web
ProdLDA	8.4±0.4	10.2±0.3	11.2±1.0	0.6±0.6
WLDA	9.4±0.2	11.0±0.5	14.6±0.4	24.2±0.7
NSTM	16.8±0.9	18.5±0.4	**18.3±0.7**	24.8±1.5
DTM	10.2±0.3	12.5±0.3	14.2±0.1	13.7±0.4
DETM	8.3±0.5	8.4±0.4	11.4±0.3	21.1±0.3
RTM	7.1±0.6	8.9±0.3	6.9±0.3	20.1±0.8
NRTM	6.9±0.4	9.2±0.4	11.6±0.4	19.9±1.7
Adjacent-Encoder	11.8±0.8	13.4±0.6	17.6±0.0	1.4±0.0
NetDTM	**18.9±0.6**	**19.4±0.5**	17.3±0.5	**29.3±0.9**
NetDTM++	16.6±0.6	**19.4±0.6**	17.8±0.3	29.0±1.2

Table 4.4: Topic coherence NPMI (results are in percentage).



(a) Topic evolution of NetDTM (b) Topic evolution of NetDTM++

Figure 4.4: Topic evolution on ML dataset.



(a) Effect of network structure (b) Effect of time (c) Effect of OT on doc classification (d) Effect of OT on topic coherence

Figure 4.5: Model analysis on ML dataset.

results, which verifies that network indeed reveals document similarities, and modeling it can improve semantic learning.

Effect of Time. We set the timestamps of all the documents to be the same, i.e., we observe the whole static document network without any evolution process. We present the classification accuracy with and without time at Fig. 4.5(b). Both NetDTM and NetDTM++ increase accuracy when considering time. NetDTM does not improve too much, since it mainly models semantics without the effect of historical events. NetDTM++ shows a significant improvement, since previously established links reveals network evolution, and disregarding time leads to worse result.

Effect of Optimal Transport. To investigate the effectiveness of optimal transport, we vary the value of η_{OT} at Eq. 4.12. For document classification at Fig. 4.5(c), as η_{OT} increases, the accuracy keeps flat or even becomes higher at the beginning, after which both models decrease the results. Moving to topic coherence at Fig. 4.5(d), we observe that OT can significantly enhance the coherence. Combining Fig. 4.5(c) and (d), we conclude that compared to the case with no OT $\eta_{OT} = 0$, an appropriate value of η_{OT} can maintain or even boost the result, while an overly high value hurts some evaluation tasks. Taking the trade-off between classification and topic coherence, we set $\eta_{OT} = 1$ to combine both OT and log-likelihood.

Effect of Timestamp Granularity. Thus far, we set annually and semimonthly as one timestamp period for HEP-TH and Web, respectively. Here, we use different periods to investigate the effect of timestamp granularity. Table 4.5 shows the result of link prediction. For HEP-TH, a short period of timestamp (quarterly and semiannually) may not observe a significant change of research topics, but brings more parameters. Thus, overfitting problem may happen and decrease the results. For Web, due to the short effective period of news articles, a long timestamp period, i.e., monthly, contains too much change of news development. A long period cannot capture the transient topic evolution, thus the results decrease.

Brief Report on Running Time. Our focus is effectiveness, not efficiency. We just briefly report running time. On the largest data Web, NetDTM takes 100 min to converge, NetDTM++

Model	HEP-TH			Web	
	Quarterly	Semianually	Annually	Semimonthly	Monthly
NetDTM	9.55±0.09	9.06±0.11	**11.42±0.15**	**16.72±0.06**	16.70±0.06
NetDTM++	11.51±0.25	10.81±0.09	**13.96±0.33**	**16.73±0.11**	16.66±0.04

Table 4.5: Time granularity on link prediction (in percentage).

takes 124 min. Experiments were done on a Tesla K80 GPU with 11441MiB.

4.5 Discussion

In this work, we propose two neural topic models for dynamic document networks, which are notable in jointly preserving dynamicity and network adjacency. By designing a time-aware optimal transport, NetDTM models each link by semantically generating content of neighbors. NetDTM++ further extends NetDTM to incorporate the effect of historical links by a Hawkes process. Experiments on several dynamic document networks covering academic literature and Web documents show the effectiveness of our models against baselines.

Chapter 5

Variational Graph Author Topic Modeling

5.1 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 [103]. Recently, Variational Graph Auto-Encoder (VGAE) [40] has presented promising ability to learn effective document representations. However, when modeling documents, we usually assume a latent topic structure [6]. 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 [6], infer topics based on plain text only, without auxiliary *authorship* or *venues*. Recently, Author Topic Models [73] are proposed for authorship and venue modeling.

Definition 5.1.1 (Documents with Authors and Venues)**.** *We are given a corpus of documents* $\mathcal{C} = \{\mathcal{D}, \mathcal{A}, \mathcal{V}, \mathcal{X}\}$ *with authors and venues.* $\mathcal{D} = \{\boldsymbol{d}_i\}$ *is a set of documents. Each document* d *contains* N_d *words in the vocabulary* $\mathcal{W}$*, i.e.,* $\boldsymbol{d} = \{w_{d,n}\}_{n=1}^{N_d} \subseteq \mathcal{W}$*. Document* d *has a sequence of* A_d *authors* $\boldsymbol{a}_d = \{a_{d,n}\}_{n=1}^{A_d} \subseteq \mathcal{A}$ *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 [90], when no edges* $\mathcal{X}$ *are observed, we induce* κ*NN edges using documents' content similarity. We include* $\mathcal{X}$ *because we will use it to construct a document graph for author topic modeling.*

Given $\mathcal{C}$, 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.

5.2 Background

Definition 5.2.1 (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* $q(\boldsymbol{Z}|\mathcal{D}, \mathcal{X}) \in \mathbb{R}^{|\mathcal{D}| \times K}$*, preserving content* $\mathcal{D}$ *and graph structure* $\mathcal{X}$*. VGAE aims to maximize the following objective.*

$$\mathcal{L} = \mathbb{E}_{q(\boldsymbol{Z}|\mathcal{D},\mathcal{X})} \log p(\mathcal{X}|\boldsymbol{Z}) - \mathcal{R}[q(\boldsymbol{Z}|\mathcal{D}, \mathcal{X})||p(\boldsymbol{Z})]. \tag{5.1}$$

Encoder $q(\mathbf{Z}|\mathcal{D},\mathcal{X})$ is variational posterior parameterized by a Graph Convolutional Network [39]. 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 chapter, we will extend VGAE as a topic model and incorporate auxiliary authorship $\mathcal{A}$ and publication venues $\mathcal{V}$.

Definition 5.2.2 (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*

$$W_\rho(\boldsymbol{u},\boldsymbol{v}) = \left(\inf_{\pi\in\Pi(\boldsymbol{u},\boldsymbol{v})}\int_{\mathbb{R}^K\times\mathbb{R}^K}||x-y||^\rho d\pi(x,y)\right)^{1/\rho}. \tag{5.2}$$

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.

5.3 Hierarchical Multi-Layered Graph

Given $\mathcal{C}$, to design graph convolution to obtain topic proportions of documents, we need to construct a document graph using $\mathcal{C}$. Below we first define a *multi-layered* graph. Then we extend it to a *hierarchical multi-layered* structure. See Fig. 5.1(a) for an overview.

5.3.1 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 ver-

tex set $\mathcal{U}$ and an edge set $\mathcal{E}$, and is associated with two mapping functions θ and ϑ. The vertex mapping function $\theta : \mathcal{U} \rightarrow \mathcal{O}$ projects each vertex $i \in \mathcal{U}$ to a specific type $o \in \mathcal{O} = \{$document, word, author, venue$\}$. Each type o corresponds to a *graph layer* containing vertices of the same type. The edge mapping function $\vartheta : \mathcal{E} \rightarrow \mathcal{T}$ 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. 5.1(a) contains 4 graph layers, black and green edges are intra- and cross-layer edges.

5.3.2 Three Word Sub-Layers

We now define word-word edges. As shown by topic model literature [17], 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 [103], 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.

$$S_{ctx}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}. \tag{5.3}$$

$p(w_i, w_j)$ 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 [52], we use Stanford CoreNLP parser [56] to extract dependency between words. For each pair of words (w_i, w_j), we calculate *syntactic co-occurrence* score by

$$S_{syn} = \frac{N_{syn}(w_i, w_j)}{N_{co\text{-}occur}(w_i, w_j)}. \tag{5.4}$$

$N_{syn}(w_i, w_j)$ is the number of times that word pair (w_i, w_j) presents syntactic dependency, which is normalized by $N_{co\text{-}occur}(w_i, w_j)$, 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 pre-trained word embeddings [68]. For each pair (w_i, w_j), we calculate *semantic co-occurrence* score.

$$S_{sem} = \cos(g(w_i), g(w_j)). \tag{5.5}$$

$g(w_i)$ 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. 5.1(a), three sub-layers of words share the same set of vertices, i.e., words, but the

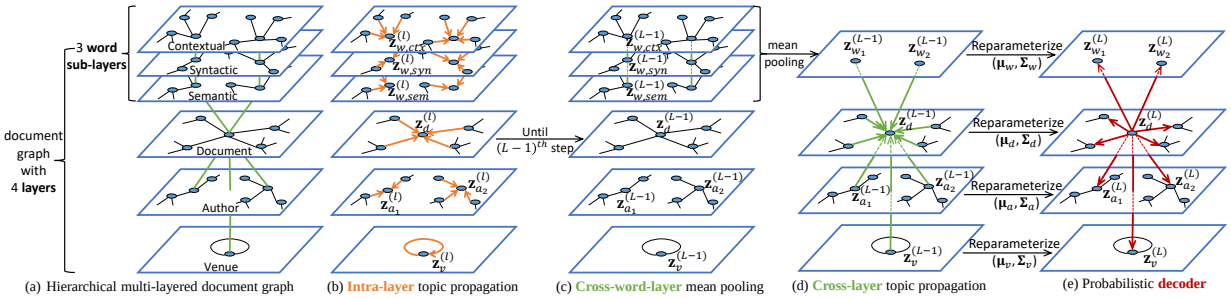


Figure 5.1: Model architecture. (a) Given a corpus with auxiliary authors and venues, we construct a hierarchical multi-layered document graph with three word relations. (b) For the first $L-1$ convolution steps, we simulate intra-layer propagation within each graph layer. (c) For the L-th convolution, 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.

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.

5.4 Model Architecture and Analysis

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

5.4.1 Generative Process

As an overview, we describe the generative process of VGATM. Following LDA, given a corpus $\mathcal{C}$, 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 $\mathcal{L}(\mathcal{C})$ is intractable, as in VGAE [40], we instead maximize its evidence lower bound below.

$$\begin{aligned}\mathcal{L} = \mathbb{E}_{q(\mathbf{Z}_\mathcal{D}, \mathbf{Z}_\mathcal{W}, \mathbf{Z}_\mathcal{A}, \mathbf{Z}_\mathcal{V})} \Big(\sum_{d \in \mathcal{D}} [\log p(\mathbf{d}|\mathbf{z}_d, \mathbf{Z}_\mathcal{W}) + \log p(\mathbf{a}_d|\mathbf{z}_d, \mathbf{Z}_\mathcal{A}) \\ + \log p(\mathbf{v}_d|\mathbf{z}_d, \mathbf{Z}_\mathcal{V}) + \lambda_{label} \log p(\mathbf{y}_d|\mathbf{z}_d)] + \sum_{d_i, d_j \in \mathcal{D}} \log p(x_{d_i,d_j}|\mathbf{z}_{d_i}, \mathbf{z}_{d_j}) \Big) \\ - \lambda_{prior} (\mathcal{R}[q(\mathbf{Z}_\mathcal{D})||p(\mathbf{Z})] + \mathcal{R}[q(\mathbf{Z}_\mathcal{W})||p(\mathbf{Z})] + \mathcal{R}[q(\mathbf{Z}_\mathcal{A})||p(\mathbf{Z})] \\ + \mathcal{R}[q(\mathbf{Z}_\mathcal{V})||p(\mathbf{Z})]).\end{aligned} \tag{5.6}$$

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 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. 5.6 concern data reconstruction, and the next two rows are divergences $\mathcal{R}$ that push variational posteriors to predefined priors as regularization. Eq. 5.6 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.

5.4.2 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.

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 $o \in \mathcal{O}$,

$$\tilde{\mathbf{z}}_i^{(l)} = \mathbf{W}_o^{(l)} \mathbf{z}_i^{(l-1)}. \tag{5.7}$$

l is the l-th convolutional step. Previous works [39] 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.

$$\alpha_{ij} = \text{softmax}\Big(\text{LeakyReLU}(\mathbf{b}_o^{(l)\top}[\tilde{\mathbf{z}}_i^{(l)}||\tilde{\mathbf{z}}_j^{(l)}])\Big), \quad j \in \mathcal{N}_o(i). \tag{5.8}$$

$\mathcal{N}_o(i)$ 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.

$$\mathbf{z}_i^{(l)} = \tanh\Big(\frac{1}{2}(\tilde{\mathbf{z}}_i^{(l)} + \sum_{j \in \mathcal{N}_o(i)} \alpha_{ij}\tilde{\mathbf{z}}_j^{(l)})\Big). \tag{5.9}$$

$\mathbf{z}_i^{(l)}$ 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,

$$\mathbf{z}_i^{(l)} = f\Big(\mathbf{z}_i^{(l-1)}, \{\mathbf{z}_j^{(l-1)} | j \in \mathcal{N}_o(i)\}\Big), \text{ where } l = 1, 2, ..., L-1. \tag{5.10}$$

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. 5.1(b) where orange arrows denote the direction of intra-layer propagation within each layer.

Cross-Layer Topic Propagation.

We now define the L-th convolutional step. As in previous works [95], 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. 5.10 for the same word w, we first unify them by a cross-word-layer mean pooling, illustrated by Fig. 5.1(c).

$$\mathbf{z}_w^{(L-1)} = \text{mean}(\mathbf{z}_{w,ctx}^{(L-1)}, \mathbf{z}_{w,syn}^{(L-1)}, \mathbf{z}_{w,sem}^{(L-1)}), \tag{5.11}$$

which is then input to the L-th step. After linear transformation at Eq. 5.7, 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.

$$\alpha_{d,w} = \text{softmax}\Big(\text{LeakyReLU}(\mathbf{b}^\top[\tilde{\mathbf{z}}_d^{(L)}||\tilde{\mathbf{z}}_w^{(L)}])\Big) \tag{5.12}$$

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

$$\tilde{\mathbf{h}}_w^{(L)} = \sum_w \alpha_{d,w}\tilde{\mathbf{z}}_w^{(L)}, \tag{5.13}$$

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. 5.12–5.13 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. 5.12,

$$\alpha_{d,a} = \text{softmax}\Big(\delta(d,a) \times \text{LeakyReLU}(\mathbf{b}^\top[\tilde{\mathbf{z}}_d^{(L)}||\tilde{\mathbf{z}}_a^{(L)}])\Big). \tag{5.14}$$

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

$$\delta(d,a) = (1/2)^{s(d,a)-1}. \tag{5.15}$$

$s(d, a)$ 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. 5.15 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. 5.1(d)). η controls the importance of cross-layer propagation.

$$\boldsymbol{\mu}_d = (1 - \eta) \times \frac{1}{2}(\tilde{\mathbf{z}}_d^{(L)} + \tilde{\mathbf{h}}_d^{(L)}) + \eta \times \text{mean}(\tilde{\mathbf{h}}_w^{(L)}, \tilde{\mathbf{h}}_a^{(L)}, \tilde{\mathbf{h}}_v^{(L)}) \tag{5.16}$$

Since we aim to output both mean and covariance from the final convolutional step, we repeat Eq. 5.11–5.16 twice and obtain $\boldsymbol{\mu}_d$ and $\boldsymbol{\Sigma}_d$ 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 [38]. For clarity, we omit superscript (L).

$$\mathbf{z}_d = \mathbf{z}_d^{(L)} = \boldsymbol{\mu}_d + (\boldsymbol{\Sigma}_d)^{1/2}\boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{5.17}$$

We will analyze the alternatives of Gaussian at Sec. 5.4.3. 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. 5.11 for cross-word-layer mean pooling and obtain $\mathbf{z}_w^{(L-1)}$, which is then input to an intra-layer convolu-

tion at Eq. 5.10.

$$\begin{aligned}\boldsymbol{\mu}_w &= f_\mu\Big(\mathbf{z}_w^{(L-1)}, \{\mathbf{z}_{w'}^{(L-1)} | w' \in \mathcal{N}_{word}(w)\}\Big) \\ \boldsymbol{\Sigma}_w &= f_\Sigma\Big(\mathbf{z}_w^{(L-1)}, \{\mathbf{z}_{w'}^{(L-1)} | w' \in \mathcal{N}_{word}(w)\}\Big).\end{aligned} \tag{5.18}$$

Finally, we apply Eq. 5.17 and obtain $\mathbf{z}_w$ for every word. For authors and venues, we simply repeat intra-layer convolutional step at Eq. 5.18 and reparameterization at Eq. 5.17 and output $\mathbf{z}_a$ and $\mathbf{z}_v$.

5.4.3 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. 5.6, which pushes $q(\mathbf{z}_i)$ to a predefined prior $p(\mathbf{z})$ using $\mathcal{R}$ as regularization. Here, we design three modeling alternatives.

KL Divergence with Gaussian Prior.

Following VGAE [40], the first design is KL divergence as $\mathcal{R}$ and isotropic Gaussian with zero mean as prior $p(\mathbf{z})$. Above reparameterization at Eq. 5.17 follows this Gaussian prior. The corresponding KL divergence is

$$\text{KL}[q(\mathbf{z}_i)||p(\mathbf{z})] = \frac{1}{2}(\text{tr}(\boldsymbol{\Sigma}_i) + \boldsymbol{\mu}_i^\top \boldsymbol{\mu}_i - \log|\boldsymbol{\Sigma}_i| - K). \tag{5.19}$$

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

KL Divergence with Dirichlet Prior.

Inspired by the success of Dirichlet prior in LDA [6], which improves topic quality, we analyze Dirichlet prior as an alternative of Gaussian. We follow [76] and evaluate Dirichlet posterior

$q(\mathbf{z}_i)$ by Laplace approximation.

$$q(\mathbf{z}_i) = \text{softmax}(\boldsymbol{\mu}_i + \boldsymbol{\Sigma}_i^{1/2}\boldsymbol{\epsilon}), \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{5.20}$$

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

$$\mu_k = \log \alpha_k - \frac{1}{K}\sum_{k'} \log \alpha_{k'}, \quad \Sigma_{kk} = \frac{1}{\alpha_k}(1 - \frac{2}{K}) + \frac{1}{K^2}\sum_{k'}\frac{1}{\alpha_{k'}} \tag{5.21}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix. After obtaining $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, we use Eq. 5.20 to get approximated Dirichlet prior $p(\mathbf{z})$. KL divergence is

$$\text{KL}[q(\mathbf{z}_i)||p(\mathbf{z})] = \tfrac{1}{2}\Big(\text{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_i) + (\boldsymbol{\mu} - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_i) + \log \tfrac{|\boldsymbol{\Sigma}|}{|\boldsymbol{\Sigma}_i|} - K\Big). \tag{5.22}$$

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 [63], 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 $\boldsymbol{\Sigma}_i$ as Gaussian variational posterior. We measure its distance with Gaussian prior.

Theorem 1. *Let $p(\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $q(\mathbf{z}_i) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ be two Gaussian distributions. Their*

2-Wasserstein distance is [97]

$$W_2[p(\boldsymbol{z}), q(\boldsymbol{z}_i)] = ||\boldsymbol{\mu} - \boldsymbol{\mu}_i||_2^2 + tr(\boldsymbol{\Sigma} + \boldsymbol{\Sigma}_i - 2(\boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{\Sigma}_i\boldsymbol{\Sigma}^{\frac{1}{2}})^{\frac{1}{2}}). \tag{5.23}$$

Wasserstein distance between two Gaussians has an analytical solution. Specifically, in our model the covariance of Gaussian prior and variational posterior is diagonal, $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma}^2)$ and $\boldsymbol{\Sigma}_i = \text{diag}(\boldsymbol{\sigma}_i^2)$, Eq. 5.23 can be simplified as a symmetric form

$$W_2[p(\mathbf{z}), q(\mathbf{z}_i)] = ||\boldsymbol{\mu} - \boldsymbol{\mu}_i||_2^2 + ||\boldsymbol{\sigma}^2 - \boldsymbol{\sigma}_i^2||_2^2. \tag{5.24}$$

We will examine the effect of these three modeling alternatives.

5.4.4 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. 5.6.

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. 5.6 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 [61] to replace it.

$$\sum_{w:d_w=1} [\log \phi(\mathbf{z}_d^\top \mathbf{z}_w) + \sum_{m=1}^{M} \mathbb{E}_{w' \sim p_n(w)} \log \phi(-\mathbf{z}_d^\top \mathbf{z}_{w'})] \tag{5.25}$$

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. 5.25) 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. 5.1(e) by red arrows.

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

$$\hat{\mathbf{y}}_d = \text{softmax}(f_{\text{MLP}}(\mathbf{z}_d)), \quad \log p(\mathbf{y}_d|\mathbf{z}_d) = \sum_n y_{d,n} \log \hat{y}_{d,n}. \tag{5.26}$$

$f_{\text{MLP}}(\cdot)$ 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. 5.6 until convergence.

5.5 Experiments

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

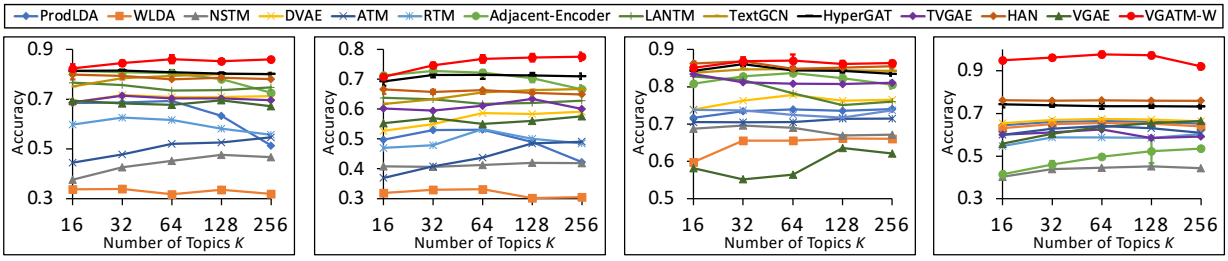
Datasets. We use six datasets at Table 5.1. Cora [58] is a corpus of papers with abstract as content and citations as doc-doc edges. Each paper has a sequence of authors. We extracted two independent datasets, Machine Learning (ML) and Programming Language (PL). Besides, we used two more datasets, HEP-TH [46] and Aminer [78] 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 [45] is a Web page hyperlink network. Each page is a news article and associated with an author.

Baselines. We consider 5 categories of baselines. *i*) **Topic models for plain text**, ProdLDA

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

Table 5.1: 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.



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

[76], WLDA [63], NSTM [118], and DVAE [8]. 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 [73] where topics of a document are the average of its authors'. *iii*) **Topic models for document graphs**, RTM [10], Adjacent-Encoder [107], and LANTM [90]. 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 [103], HyperGAT [20], TVGAE [95]. 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 [89] as *supervised* and VGAE [40] 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.

5.5.1 Quantitative Evaluation

Document Classification. Following LDA [6], 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.

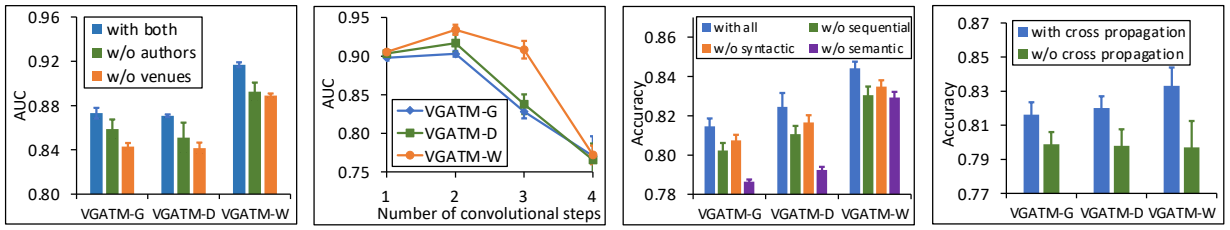
Supervised Training. Labels are involved for supervised training. We compare to all baselines. Supervised baselines output predicted labels for documents, which are then compared with ground-truth labels. For completeness, we also compare to unsupervised baselines, which output topic proportions without label prediction. We follow [107] and train an external kNN classifier ($k = 5$) using the output topics of training documents and predict labels of test documents. Fig. 5.2 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 5.2 shows the accuracy at 64 topics.

Analysis. For both classification tasks, the best baselines are Adjacent-Encoder, LANTM,

Table 5.2: 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



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

Figure 5.3: Ablation analysis of our models.

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 5.2, which verifies that Wasserstein is a promising alternative of KL. Dirichlet prior performs better than Gaussian. As verified by previous work [8], Dirichlet encourages topics to be sparser than Gaussian and achieves a lower reconstruction error, thus improving topic quality.

Link Prediction. Edges reveal semantic similarity between documents. As in RTM [10], we conduct link prediction to evaluate topic quality. As in [107], 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.

Table 5.3: 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					
		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
	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
	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
	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
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
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
	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
	LANTM	80.6±1.2	75.4±0.7	84.9±1.1	86.1±0.3	N.A.	N.A.
Text classification models	TextGCN	81.3±0.3	75.4±0.4	81.1±0.1	N.A.	N.A.	N.A.
(they are *supervised* and cannot run on	HyperGAT	83.1±0.5	79.7±0.5	87.1±0.3	N.A.	90.0±0.0	N.A.
HEP-TH and Web with no observed labels)	TVGAE	79.1±0.7	74.7±1.0	88.2±1.0	N.A.	85.3±0.6	N.A.
Graph embedding models	HAN	77.0±0.7	73.1±0.4	84.7±1.0	N.A.	93.2±0.1	N.A.
(HAN is *supervised*, cannot run without labels)	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
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
	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
	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**

Category	Model	Doc-Author Link Prediction					
		ML	PL	COVID	HEP-TH	Aminer	Web
Models for plain text	ProdLDA	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	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	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	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	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	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	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	86.1±0.9	87.8±0.8	71.0±1.5	85.7±0.3	N.A.	N.A.
Text classification models	TextGCN	56.8±0.7	50.4±1.6	47.7±5.2	N.A.	N.A.	N.A.
(they are *supervised* and cannot run on	HyperGAT	50.0±0.8	49.6±0.7	61.8±3.1	N.A.	49.1±0.2	N.A.
HEP-TH and Web with no observed labels)	TVGAE	65.0±0.9	65.4±0.9	72.8±1.5	N.A.	70.6±0.7	N.A.
Graph embedding models	HAN	73.0±1.4	72.2±2.2	79.2±1.1	N.A.	71.3±1.1	N.A.
(HAN is *supervised*, cannot run without labels)	VGAE	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	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	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	**93.0±0.3**	**93.8±0.5**	**79.5±1.2**	**91.2±0.3**	**74.1±0.3**	**77.3±0.0**

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 [107], 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 [90]. Table 5.3 (upper) 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

Table 5.4: Topic coherence NPMI at $K = 64$.

Category	Model	Topic Coherence NPMI					
		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
	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
	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
	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
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
Models with document graph	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
(LANTM cannot run on large dataset	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
Aminer and Web even on 256GB machine)	LANTM	9.9±1.2	9.8±0.7	8.6±0.3	10.4±1.5	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.
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
	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**
	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

Table 5.5: Perplexity at $K = 64$.

Category	Model	Perplexity					
		ML	PL	COVID	HEP-TH	Aminer	Web
Models for plain text	ProdLDA	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	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	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	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	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	RTM	8.07±0.01	7.93±0.01	8.98±0.04	8.04±0.00	8.89±0.01	10.28±0.19
(LANTM cannot run on large dataset	Adjacent-Encoder	7.41±0.01	7.34±0.13	6.96±0.00	7.45±0.19	8.71±0.02	8.26±0.01
Aminer and Web even on 256GB machine)	LANTM	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	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	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	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	**5.23±0.15**	**5.13±0.30**	**6.55±0.20**	**4.94±0.06**	**5.75±0.28**	5.60±0.41

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 5.3 (lower) 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.

5.5.2 Topic Analysis

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

Table 5.6: 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

distribution of a topic over the words, and the highest values on that column are the key words of that topic. As in ProdLDA [76], we evaluate the coherence of key words by an external corpus, Google Web 1T 5-gram Version 1 [22], with NPMI as metric. Table 5.4 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.

Perplexity. Following LDA [6], 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 5.5 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.

Interpretability. To understand what topics our models capture, we randomly select two topics for each variant and present top-5 words on COVID at Table 5.6. VGATM-G captures *children's health* and *manufacture depression*. VGATM-D reveals *retirement* and *sports*. VGATM-W shows *studying at home* and *confirmed cases*.

5.5.3 Model Analysis

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. 5.3(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.

Number of Convolutional Steps. We analyze the performance of different convolutional steps L at Fig. 5.3(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.

Three Word Co-occurrence Relations. Here we test the effectiveness of three word relations by removing each one from the complete models. Fig. 5.3(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.

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. 5.16 and maintain intra-layer propagation only. Fig. 5.3(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.

5.6 Discussion

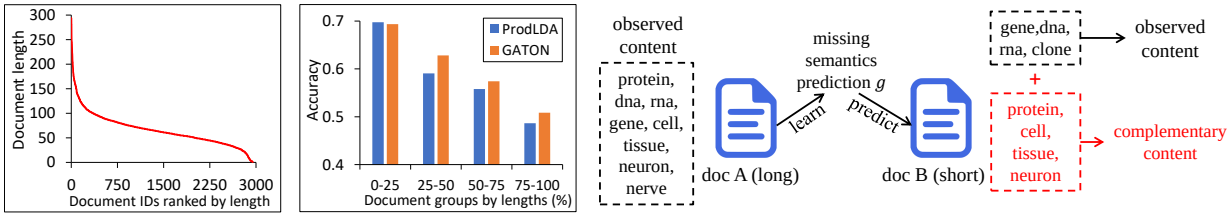
We propose Variational Graph Author Topic Model, which flexibly works under supervised and unsupervised settings. To model authors, venues, and three word relations, we design a hierarchical multi-layered document graph and propose three alternatives of divergence. Experiments verify the effectiveness of various components of our models ablatively, as well as the holistic model's outperformance over baselines.

Chapter 6

Meta-Complementing the Semantics of Short Texts in Neural Topic Models

6.1 Introduction

Much of the data on the Web can be represented as text documents. Topic models help to understand the main themes within documents, i.e., each document is represented by a topic distribution, and each topic is interpreted by its key words. The quality of topic distribution of each document depends on sufficient word co-occurrences. However, many real-world corpora contain documents of *variable lengths*. Academic papers vary from journal manuscripts to conference papers to extended abstracts. News articles could be headlines, short or full articles, or detailed commentaries. Fig. 6.1(a) illustrates an academic paper corpus where the distribution of document lengths exhibits a long-tail distribution. Despite variable lengths (with different degrees of sufficiency of word co-occurrences), existing works, e.g., ProdLDA [76] and GATON [100], treat documents uniformly, resulting in inferior topic quality for short texts. Evidentially, Fig. 6.1(b) presents document classification accuracy on four subsets of corpus with descending lengths. Accuracies gradually drop as the length decreases. The inferior topic quality of short texts limits the overall performance of a topic model.



(a) Document lengths (b) Classification accuracy (c) Semantics transfer and complement

Figure 6.1: Illustration of (a) a paper corpus with various-length documents, (b) classification accuracy on four subsets of the corpus by descending length, and (c) semantic transfer and complement.

Problem. We are given a corpus of N documents $\mathcal{D} = \{\mathbf{d}_i\}_{i=1}^N$. Each document $\mathbf{d} \in \mathbb{R}^{|\mathcal{V}|}$ is a vector in the vocabulary space $\mathcal{V}$. $l_d = \sum_{w \in \mathcal{V}} d_w$ is the length of document d where d_w is the word count of w in d. When word embeddings are available, we have $\mathcal{H} = \{\mathbf{h}_w\}_{w \in \mathcal{V}}$ where $\mathbf{h}_w$ is the embedding of word w. Documents may link to others in a document network $\mathcal{G} = \{\mathcal{D}, \mathcal{E}\}$, with documents $\mathcal{D}$ and network connectivity $\mathcal{E} = \{e_{ij}\}_{i,j=1}^N$. If document d links to d', $e_{d,d'} = 1$, otherwise $e_{d,d'} = 0$. $\mathcal{N}(d)$ is the set of neighbors of d. We consider an undirected network, $e_{d,d'} = e_{d',d}$. Corpus $\mathcal{D}$ contains documents of variable lengths $\{l_i\}_{i=1}^N$. We introduce a hyperparameter $\mathscr{L}$ as the threshold where short documents are those with fewer observed words, $\mathcal{D}_{\text{short}} = \{\mathbf{d}_i | l_i < \mathscr{L}\}$, and long documents are defined symmetrically, $\mathcal{D}_{\text{long}} = \{\mathbf{d}_i | l_i \geq \mathscr{L}\}$. We consider $\mathscr{L}$ as a predefined hyperparameter and leave other designs as future work.

Given a variable-length corpus $\mathcal{D}$ (as well as word embeddings $\mathcal{H}$ and network $\mathcal{E}$ if observed) as input, we aim to output topic distributions for documents where the topic quality of short documents is improved, without hurting long documents. Note that our goal is not to allow short texts to reach the performance of long documents, but to improve short text topic modeling as much as possible.

6.2 Background

Meta-Learning. Meta-learning [23] optimizes globally shared parameters, a.k.a. prior knowledge, over meta-training tasks, so as to rapidly adapt the model to previously unseen meta-testing tasks with only a few observed data. Since topic models generally learn topics by a content generative process, here we consider generating observed words for a document d as a task $\mathcal{T}_d$. A meta-training task $\mathcal{T}_d$ corresponds to a training document d and consists of a support set and a query set, $\mathcal{T}_d = \{\mathcal{S}_d, \mathcal{Q}_d\}$. Each set contains randomly sampled words from document d, such that support words and query words are mutually exclusive, $\mathcal{S}_d \cap \mathcal{Q}_d = \emptyset$ and $\mathcal{S}_d \cup \mathcal{Q}_d = \mathbf{d}$. *Meta-training* has two steps:

1. Local update. Given a topic model f_θ with parameter θ, f_θ is first updated from the globally shared parameter θ to document-specific local parameter θ_d w.r.t. loss on d's support words $\mathcal{S}_d$.

$$\theta_d = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{S}_d) \quad \text{where} \quad \mathcal{T}_d = \{\mathcal{S}_d, \mathcal{Q}_d\} \in \mathcal{T}_{\text{tr}}. \tag{6.1}$$

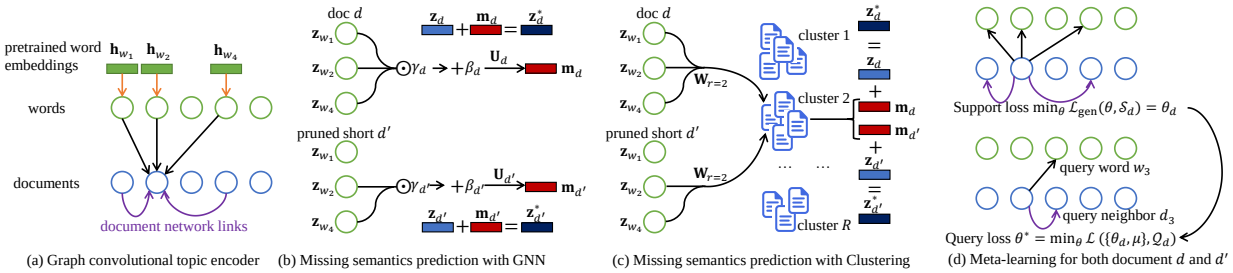
α is meta-learning rate, ∇ is gradient, $\mathcal{L}$ is loss function, $\mathcal{T}_{\text{tr}}$ is a set of meta-training tasks (documents).

2. Global update. After obtaining θ_d for each document d, we compute the loss on query words $\mathcal{L}(\theta_d, \mathcal{Q}_d)$. Together with other training tasks, we optimize the globally shared parameter θ.

$$\theta^* = \min_\theta \sum_{\mathcal{T}_d \in \mathcal{T}_{\text{tr}}} \mathcal{L}(\theta_d, \mathcal{Q}_d) = \min_\theta \sum_{\mathcal{T}_d \in \mathcal{T}_{\text{tr}}} \mathcal{L}(\theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{S}_d), \mathcal{Q}_d). \tag{6.2}$$

θ^* is the new globally shared parameter and will replace θ at Eq. 6.1–6.2 for the next iteration. After convergence, the final global parameter θ^* can easily be adapted to meta-testing tasks.

Meta-testing tasks $\mathcal{T}_{\text{te}}$ are unseen test documents. All the observed words are support words, $\mathcal{T}_d = \mathcal{S}_d = \mathbf{d}$. During meta-testing, topic model f_{θ^*} with optimized global parameter θ^* is updated w.r.t. $\mathcal{S}_d$ by Eq. 6.1 and obtain θ_d^*. The topic distribution of testing document is inferred by $\mathbf{z}_d = f_{\theta_d^*}(\mathbf{d})$.



(a) Graph convolutional topic encoder (b) Missing semantics prediction with GNN (c) Missing semantics prediction with Clustering (d) Meta-learning for both document d and d'

Figure 6.2: Model architecture of Meta-Complementing Topic Model, MCTM.

6.3 Model Architecture and Analysis

We introduce **M**eta-**C**omplement **T**opic-**M**odel (MCTM) at Fig. 6.2. Below we elaborate three components, graph convolutional encoder, missing semantics prediction, and meta-learning optimization.

6.3.1 Graph Convolutional Topic Encoding

We follow GATON [100] and first present a graph convolutional encoder f_θ (Fig. 6.2(a)), which projects documents to K-dimensional topic distributions. We defer the discussion on short text modeling to the following subsections. Given a corpus $\mathcal{D}$, considering documents and words as vertices, we construct a bipartite graph, the links represent word occurrences in the documents. Since documents and words preserve heterogeneous feature spaces, we project them to the same topic space by learnable matrix,

$$\tilde{\mathbf{z}}_d^{(l+1)} = \mathbf{W}_1^{(l+1)} \mathbf{z}_d^{(l)}, \qquad \tilde{\mathbf{z}}_w^{(l+1)} = \mathbf{W}_2^{(l+1)} \mathbf{z}_w^{(l)}, \quad \text{where} \quad d \in \mathcal{D}, w \in \mathcal{V}. \tag{6.3}$$

l is the l-th convolutional layer. $\mathbf{z}_d^{(0)}$ and $\mathbf{z}_w^{(0)}$ are inputs, i.e., Bag-of-Words and one-hot, respectively.

To differentiate the importance of words, we evaluate attention between document d and its

observed support words at Eq. 6.4 and aggregate words at Eq. 6.5 where $[\cdot||\cdot]$ is concatenation.

$$a_{d,w} = \text{softmax}\Big(\tanh(\mathbf{b}_1^{(l+1)\top}[\tilde{\mathbf{z}}_d^{(l+1)}||\tilde{\mathbf{z}}_w^{(l+1)}])\Big) \quad \text{where} \quad w \in \mathcal{S}_d \tag{6.4}$$

$$\mathbf{z}_d^{(l+1)} = \tanh\Big(\frac{1}{2}(\tilde{\mathbf{z}}_d^{(l+1)} + \sum_{w \in \mathcal{S}_d} a_{d,w}\tilde{\mathbf{z}}_w^{(l+1)})\Big). \tag{6.5}$$

Symmetrically, we modify Eq. 6.4 and obtain $a_{w,d}$, i.e., the attention between word w and documents it appears in. After symmetric aggregation at Eq. 6.5, we obtain $\mathbf{z}_w^{(l+1)}$ for word w. So far, we complete the convolution from l-th to $(l+1)$-th layer. For simplicity, we summarize aggregation at Eq. 6.3–6.5 by

$$\mathbf{z}_d^{(l+1)} = \text{AGG}(\mathbf{W}_1^{(l+1)}\mathbf{z}_d^{(l)}, \mathbf{W}_2^{(l+1)}\mathbf{z}_w^{(l)}|w \in \mathcal{S}_d), \quad \mathbf{z}_w^{(l+1)} = \text{AGG}(\mathbf{W}_2^{(l+1)}\mathbf{z}_w^{(l)}, \mathbf{W}_1^{(l+1)}\mathbf{z}_d^{(l)}|\forall d : w \in \mathcal{S}_d). \tag{6.6}$$

We repeat Eq. 6.6 for maximum L layers and obtain K-dimensional topics $\mathbf{z}_d = \mathbf{z}_d^{(L)}$ for document d and $\mathbf{z}_w = \mathbf{z}_w^{(L)}$ for word w. The complete encoder is Eq. 6.7. θ is the set of all encoding parameters.

$$\mathbf{z}_d, \mathbf{z}_w = f_\theta(\mathbf{z}_d^{(l=0)}, \mathbf{z}_w^{(l=0)}|d \in \mathcal{D}, w \in \mathcal{V}). \tag{6.7}$$

6.3.2 Missing Semantics Prediction with Contrastive Learning

A short document with few words leaves some content poorly described, resulting in incomplete topic distribution $\mathbf{z}_d$. As a toy example, document B at Fig. 6.1(c) contains limited words, e.g., gene and clone, and leaves other contents, e.g., protein and cell, uncovered. We aim to complement the topics of short documents. For a document d, regardless of long or short, we complement its semantics by

$$\mathbf{z}_d^* = \mathbf{z}_d + \mathbf{m}_d. \tag{6.8}$$

We name $\mathbf{m}_d \in \mathbb{R}^K$ *missing semantics* of document d. If d is a long document with complete semantics, $\mathbf{m}_d$ is a zero vector. A function g_μ predicts missing semantics at Eq. 6.9 with topic

distributions of d and its support words as inputs. We will elaborate the design of function g_μ shortly.

$$\mathbf{m}_d = g_\mu(\mathbf{z}_d, \mathbf{z}_w | w \in \mathcal{S}_d). \tag{6.9}$$

Contrastive Learning. Long documents contain relatively more sufficient word co-occurrences than short documents. Thus, we learn missing semantics prediction function g_μ on long documents, and then transfer the learned semantic knowledge to complement short documents. On one hand, a long document d with enough content does not need semantic complement. Thus we have below constraint

$$\mathbf{m}_d \to \mathbf{0} \quad \text{where} \quad d \in \mathcal{D}_{\text{long}}. \tag{6.10}$$

On the other hand, although we aim to transfer the semantic knowledge from long to short documents, there does not exist a one-to-one correspondence in corpus $\mathcal{D}$. As a result, we may transfer semantics of a long document (e.g., machine learning concepts) to a short one describing completely distinct content (e.g., biology). To overcome this limitation, we introduce another constraint with contrastive objective. For a long document d with support words $\mathcal{S}_d$, we randomly hide a proportion of words to mimic a short document, denoted as d' with remaining observed words $\mathcal{S}_{d'} \subset \mathcal{S}_d$ and length $l_{d'} < \mathscr{L}$. For both long text d and its short version d', we predict their missing semantics by Eq. 6.9, and obtain $\mathbf{m}_d$ and $\mathbf{m}_{d'}$, respectively. $\mathbf{m}_{d'}$ should complement the previously hidden semantics, i.e., $\mathcal{S}_d - \mathcal{S}_{d'}$,

$$\mathbf{z}_{d'} + \mathbf{m}_{d'} \to \mathbf{z}_d + \mathbf{m}_d \quad \Rightarrow \quad \mathbf{z}^*_{d'} \to \mathbf{z}^*_d \quad \text{where} \quad d \in \mathcal{D}_{\text{long}}. \tag{6.11}$$

Together with above Eq. 6.10, which forces $\mathbf{z}^*_d = \mathbf{z}_d + \mathbf{m}_d$ to approach $\mathbf{z}_d$, Eq. 6.11 actually has

$$\mathbf{z}_{d'} + \mathbf{m}_{d'} \to \mathbf{z}_d \quad \Rightarrow \quad \mathbf{m}_{d'} \to \mathbf{z}_d - \mathbf{z}_{d'} \quad \text{where} \quad d \in \mathcal{D}_{\text{long}}. \tag{6.12}$$

To summarize, we arrive at the following constraint loss.

$$\mathcal{L}_{\text{con}}(d) = -\log \sigma(\cos(\mathbf{m}_{d'}, \mathbf{z}_d - \mathbf{z}_{d'})) \quad \text{where} \quad d \in \mathcal{D}_{\text{long}}. \tag{6.13}$$

$\sigma(x) = \frac{1}{1+\exp(-x)}$ is sigmoid function, and $\cos(\cdot, \cdot)$ is cosine similarity. We here use cosine similarity, mainly due to its superior performance on our datasets. Besides cosine, other similarity metrics, such as inner product and Euclidean distance, are also possible, depending on different datasets.

Missing Semantics Prediction. We now define the function of missing semantics prediction g_μ. Given topic distributions of a document and its observed support words as input, a desirable function should aggregate them and output a single missing semantics vector. We propose two alternatives.

1. *GNN function.* The first is to implement g_μ using a graph neural network, see Fig. 6.2(b).

$$\mathbf{m}_d = g_\mu(\mathbf{z}_d, \mathbf{z}_w | w \in \mathcal{S}_d) = \text{AGG}(\mathbf{U}_1\mathbf{z}_d, \mathbf{U}_2\mathbf{z}_w | w \in \mathcal{S}_d). \tag{6.14}$$

A corpus $\mathcal{D}$ contains documents with diverse themes. For example, some documents discuss machine learning, while others describe biology. However, the assumption of corpus-level shared parameter $\mathbf{U}_1$ and $\mathbf{U}_2$ is not flexible to model diverse documents for missing semantics prediction, since different documents may have distinct optimal parameters, which are sometimes even in opposing direction. As a result, the predicted missing semantics $\mathbf{m}_d$ centers its mass around the most frequent topics and leaves other distinct topics uncovered. We seek to *personalize* $\mathbf{U}_1$ and $\mathbf{U}_2$ for each document d to recover the distinct missing semantics. Formally, we introduce a function ϕ, which transforms $\mathbf{U}_1$ and $\mathbf{U}_2$ to document-specific parameters $\mathbf{U}_{d,1}$ and $\mathbf{U}_{d,2}$ by scaling and shifting. Taking $\mathbf{U}_1$ as example,

$$\mathbf{U}_{d,1} = \phi(\mathbf{U}_1, \mathbf{z}_d, \mathbf{z}_w | w \in \mathcal{S}_d) = \mathbf{U}_1 \odot \left[(\boldsymbol{\gamma}_d + \mathbf{1})_{\times K}\right] + \left[(\boldsymbol{\beta}_d)_{\times K}\right]. \tag{6.15}$$

$\boldsymbol{\gamma}_d$ and $\boldsymbol{\beta}_d$ are document-specific vectors for scaling and shifting shared parameter $\mathbf{U}_1$. $\odot$ is element-wise product. $[(\mathbf{x})_{\times K}]$ is a matrix with K identical column vector $\mathbf{x}$. $\mathbf{1}$ is a vector of ones, ensuring the scaling matrix centers around one. $\mathbf{U}_{d,2}$ is similarly defined. Eq. 6.15 allows each document d to have its own parameters, while all documents still share the common knowledge. Similar documents scale and shift $\mathbf{U}_1$ and $\mathbf{U}_2$ to similar directions. Different documents push them to distinct directions.

We define scaling $\boldsymbol{\gamma}_d$ and shifting $\boldsymbol{\beta}_d$, parameterized by topics of document d and its support words.

$$\boldsymbol{\gamma}_d = \tanh(\mathbf{W}_\gamma \mathbf{z}_d + \mathbf{W}'_\gamma \bar{\mathbf{z}}_{\mathcal{S}_d}), \quad \boldsymbol{\beta}_d = \tanh(\mathbf{W}_\beta \mathbf{z}_d + \mathbf{W}'_\beta \bar{\mathbf{z}}_{\mathcal{S}_d}). \tag{6.16}$$

$\bar{\mathbf{z}}_{\mathcal{S}_d} = \frac{1}{|\mathcal{S}_d|} \sum_{w \in \mathcal{S}_d} \mathbf{z}_w$ is the average of d's words. In summary, we use Eq. 6.14 to predict d's missing semantics, except that shared parameters $\mathbf{U}_1$ and $\mathbf{U}_2$ are replaced by d-specific ones, $\mathbf{U}_{d,1}$ and $\mathbf{U}_{d,2}$.

2. *Clustering function.* We propose an alternative method by semantic clustering to recover distinct missing semantics, Fig. 6.2(c). Documents with similar content fall into related clusters, while unique documents belong to different ones. If we assign each cluster a set of parameters for missing semantics prediction, similar documents would recover their own distinct topics. Specifically, we introduce R centroids $\{\mathbf{c}_r\}_{r=1}^R$, each corresponding to one cluster. Given topic distributions of document d and its support words, we first evaluate the assignment probability between document d and each cluster by

$$\Pr(r) = \text{softmax}(-\frac{1}{2}||\mathbf{h}_0 - \mathbf{c}_r||_2^2) = \frac{\exp(-\frac{1}{2}||\mathbf{h}_0 - \mathbf{c}_r||_2^2)}{\sum_{r'=1}^R \exp(-\frac{1}{2}||\mathbf{h}_0 - \mathbf{c}_{r'}||_2^2)}. \tag{6.17}$$

$\mathbf{h}_0 = [\mathbf{z}_d||\bar{\mathbf{z}}_{\mathcal{S}_d}]$ is the concatenation of $\mathbf{z}_d$ and $\bar{\mathbf{z}}_{\mathcal{S}_d}$. We then assign parameters to each cluster,

$$\mathbf{m}_d = \mathbf{h}_1 = \sum_{r=1}^R \Pr(r) \times \text{ReLU}(\mathbf{W}_r \mathbf{h}_0 + \mathbf{b}_r). \tag{6.18}$$

Therefore, related documents obtain similar clustering probabilities and missing semantics.

Above process is a flat clustering. Our function can be extended to multiple clustering layers. Each layer s consists of $R^{(s)}$ centroids. After obtaining the output from previous layer, $\mathbf{h}_{s-1}$, we repeat Eq. 6.17–6.18 by replacing $\mathbf{h}_0$ with $\mathbf{h}_{s-1}$, and obtain the output of the current layer s, i.e., $\mathbf{h}_s$. For maximum S layers, we get $\mathbf{m}_d = \mathbf{h}_S$. We leave adaptive learning of number of clusters R as future work.

6.3.3 Probabilistic Decoding with Meta-Learning Optimization

After semantic complement, we obtain $\{\mathbf{z}_d^*\}_{d\in\mathcal{D}}$ at Sec. 6.3.2. We use $\mathbf{Z}_\mathcal{V} = \left[\mathbf{z}_{w_1}; \mathbf{z}_{w_2}; ...; \mathbf{z}_{w_{|\mathcal{V}|}}\right] \in \mathbb{R}^{K\times|\mathcal{V}|}$ to represent topic-word distribution, each column $\mathbf{z}_w$ is topic distribution of a word w, and each row is to the distribution of a topic over the vocabulary. As in previous topic models [6, 76], we generate the observed support words $\mathcal{S}_d$ by $\hat{\mathbf{d}}_{\mathcal{S}_d} = \sigma(\mathbf{Z}_\mathcal{V}\mathbf{z}_d^*)$. Compared to the ground-truth support words $\mathbf{d}_{\mathcal{S}_d}$, we follow [100] and obtain generative loss $\mathcal{L}_{\text{gen}} = ||\mathbf{d}_{\mathcal{S}_d} - \hat{\mathbf{d}}_{\mathcal{S}_d}||_2^2$. However, this loss requires inefficient computation over the whole vocabulary. We instead use negative sampling [61].

$$\mathcal{L}_{\text{gen}}(d) = \sum_{w\in\mathcal{S}_d}\left[(d_w - \hat{d}_w)^2 + \sum_{m=1}^{M}\mathbb{E}_{w'\sim\Pr_n(w)}(d_{w'} - \hat{d}_{w'})^2\right]. \tag{6.19}$$

M is the number of negative samples, and $\Pr_n(w)$ is a noise distribution over vocabulary. $\hat{d}_w = \sigma(\mathbf{z}_d^{*\top}\mathbf{z}_w)$. $d_w = 1$ if $w \in \mathcal{S}_d$, otherwise $d_w = 0$. In addition, if d is a long document $d \in \mathcal{D}_{\text{long}}$, we also created its corresponding pruned short version d' with a subset of support words $\mathcal{S}_{d'} \subset \mathcal{S}_d$ at Sec. 6.3.2. Although we do not observe the complete support words for d', its complete topic distribution $\mathbf{z}_{d'}^*$ after semantic complement at Eq. 6.8 should be able to generate the complete support words $\mathcal{S}_d$. Therefore, together with semantic complement constraint at Eq. 6.13, we

arrive at the complete loss.

$$\mathcal{L}(d) = \mathcal{L}_{\text{gen}}(d) + \mathcal{L}_{\text{con}}(d) + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}(\theta)$$
$$\text{where} \quad \mathcal{L}_{\text{gen}}(d) = \mathcal{L}_{\text{gen}}(d) + \mathbb{I}(d \in \mathcal{D}_{\text{long}})\lambda_{\text{gen}}\mathcal{L}_{\text{gen}}(d'), \quad \mathcal{L}_{\text{con}}(d) = \mathbb{I}(d \in \mathcal{D}_{\text{long}})\mathcal{L}_{\text{con}}(d). \tag{6.20}$$

$\mathcal{L}_{\text{gen}}(d)$ is generative loss, consisting of document d in the corpus $d \in \mathcal{D} = \mathcal{D}_{\text{long}} \cup \mathcal{D}_{\text{short}}$ and the corresponding pruned short version d' (if $d \in \mathcal{D}_{\text{long}}$). $\mathbb{I}(d \in \mathcal{D}_{\text{long}}) = 1$ if $d \in \mathcal{D}_{\text{long}}$, otherwise 0. $\mathcal{L}_{\text{reg}}(\theta)$ is L2 regularizer for encoding parameters θ. λ_{gen} and λ_{reg} are hyperparameters.

Optimization. Finally, with the objective of meta-learning at Sec. 6.2, we reach the optimization:

1. *Local update.* Given topic encoder f_θ defined at Sec. 6.3.1, we optimize encoding parameter θ w.r.t. the generative loss $\mathcal{L}_{\text{gen}}(d)$ on support words $\mathcal{S}_d$ by Eq. 6.1 and obtain θ_d for each document.

2. *Global update.* With encoding parameter θ_d, we compute the overall loss $\mathcal{L}(d)$ on query words $\mathcal{Q}_d$ and optimize all parameters $\Phi = \{\theta, \mu\}$, including encoder parameters θ and parameters μ of missing semantics prediction function g_μ. α_1 and α_2 are local and global learning rate, respectively.

$$\Phi^* = \min_\Phi \textstyle\sum_{d\in\mathcal{D}} \mathcal{L}(\{\theta_d, \mu\}, \mathcal{Q}_d) \quad \Rightarrow \quad \Phi^* = \Phi - \alpha_2 \nabla_\Phi \textstyle\sum_{d\in\mathcal{D}} \mathcal{L}(\theta - \alpha_1 \nabla_\theta \mathcal{L}_{\text{gen}}(\{\theta, \mu\}, \mathcal{S}_d), \mathcal{Q}_d). \tag{6.21}$$

With new parameter $\Phi^* = \{\theta^*, \mu^*\}$, we substitute it for Φ for the next iteration. In contrast to previous topic models that generate the content of given documents only, we further create the short version of long documents for semantic complement, and jointly optimize them. See supplementary materials for learning algorithm.

6.3.4 Extensions with Auxiliary Data

MCTM with Pretrained Word Embeddings. Pretrained word embeddings [61, 68] encode word similarity. As in previous works [17, 118], we incorporate them into topic-word distribution $\mathbf{Z}_{\mathcal{V}} = [z_{k,w}]$ to improve topic modeling. We introduce topic embedding $\{\mathbf{t}_k\}_{k=1}^{K}$ and evaluate cosine similarity between topic k and word w by $\cos(\mathbf{t}_k, \mathbf{h}_w)$. We then combine it with topic-word distribution by $z'_{k,w} = \frac{1}{2}(z_{k,w} + \cos(\mathbf{t}_k, \mathbf{h}_w))$ and obtain a new topic-word distribution $\mathbf{Z}'_{\mathcal{V}} = [z'_{k,w}]$ for decoding.

MCTM with Document Network. A document network (e.g., citation network) reveals semantic similarities between connected documents (cited papers discuss related research). A document's degree or number of links exhibits a long-tail distribution. Some link to many neighbors, others to a few. Previously focus was on *textual* semantic complement. We extend MCTM to model *structural* semantic complement for link-scarce documents. We consider generating both observed words and neighbors as a task $\mathcal{T}_d$. As for words, we split the neighbors $\mathcal{N}(d)$ of a document d into support and query neighbors $\mathcal{T}_d = \{\mathcal{S}_d, \mathcal{S}_{\mathcal{N}(d)}, \mathcal{Q}_d, \mathcal{Q}_{\mathcal{N}(d)}\}$, $\mathcal{S}_d$ and $\mathcal{S}_{\mathcal{N}(d)}$ denote support words and neighbors, respectively, and ditto for query sets. We correspondingly extend three modeling components.

1. *Encoding.* Previously, we inferred topic distribution $\mathbf{z}_d$ of document d by its textual words at Eq. 6.6. Here, we extend this process by designing a structural convolutional module.

$$\boldsymbol{\kappa}_d^{(l+1)} = \text{AGG}(\mathbf{W}_3^{(l+1)}\boldsymbol{\kappa}_d^{(l)}, \mathbf{W}_3^{(l+1)}\boldsymbol{\kappa}_{d'}^{(l)} | d' \in \mathcal{S}_{\mathcal{N}(d)}). \tag{6.22}$$

The topic distribution from encoder f_θ is $\mathbf{z}_d := \frac{1}{2}(\mathbf{z}_d + \boldsymbol{\kappa}_d)$, with both texts $\mathcal{S}_d$ and structure $\mathcal{S}_{\mathcal{N}(d)}$.

2. *Semantics complement.* For a long document d, in addition to randomly hiding some words, we also drop some neighbors and create a pruned version d'. Now the missing semantics $\mathbf{m}_d$ should contain both textual and structural information. For GNN function g_μ, we extend Eq.

Table 6.1: Dataset statistics.

Name	#Documents	#Links	Vocabulary	#Labels	Avg. #words/doc	Std.Dev. of #words/doc
ML	2,947	8,146	5,814	7	66.7	34.0
PL	2,449	7,274	5,066	9	66.0	36.9
HEP-TH	20,151	234,193	5,001	N.A.	48.4	22.9
Web	116,544	309,499	5,021	N.A.	34.1	70.0

6.14 by

$$\mathbf{m}_d = g_\mu(\mathbf{z}_d, \mathbf{z}_w, \mathbf{z}_{d'} | w \in \mathcal{S}_d, d' \in \mathcal{S}_{\mathcal{N}(d)}) = \text{AGG}(\mathbf{U}_{d,1}\mathbf{z}_d, \mathbf{U}_{d,2}\mathbf{z}_w, \mathbf{U}_{d,3}\mathbf{z}_{d'} | w \in \mathcal{S}_d, d' \in \mathcal{S}_{\mathcal{N}(d)}). \tag{6.23}$$

Scaling has extra input, $\boldsymbol{\gamma}_d = \tanh(\mathbf{W}_\gamma \mathbf{z}_d + \mathbf{W}'_\gamma \bar{\mathbf{z}}_{\mathcal{S}_d} + \mathbf{W}''_\gamma \bar{\mathbf{z}}_{\mathcal{S}_{\mathcal{N}(d)}})$, ditto for shifting. $\bar{\mathbf{z}}_{\mathcal{S}_{\mathcal{N}(d)}} = \frac{1}{|\mathcal{S}_{\mathcal{N}(d)}|} \sum_{d' \in \mathcal{S}_{\mathcal{N}(d)}} \mathbf{z}_{d'}$. For Clustering g_μ, we extend input by $\mathbf{h}_0 = [\mathbf{z}_d || \bar{\mathbf{z}}_{\mathcal{S}_d} || \bar{\mathbf{z}}_{\mathcal{S}_{\mathcal{N}(d)}}]$.

3. *Decoding with meta-learning.* In addition to generating support words using complemented $\mathbf{z}^*_d$, we also generate support neighbors. The generative loss is similar to Eq. 6.19 except that *i*) we replace d_w with $e_{d,d'}$, the ground-truth link between d and d'; *ii*) $\hat{e}_{d,d'} = \sigma(\mathbf{z}^{*\top}_d \mathbf{z}^*_{d'})$. Finally, meta-learning learns how to accurately predict missing semantics $\mathbf{m}_d$ for both textual and structural complement.

6.4 Experiments

The goal of experiments is to evaluate if our model MCTM can improve short text topic modeling through evaluative tasks, e.g., document classification, link prediction, topic analysis.

Datasets. Since our model is flexible to incorporate auxiliary data, we rely on four datasets with textual documents, auxiliary word embeddings, and auxiliary network links for experiments. Cora [58] is corpus of academic papers with citations as links. We created two independent datasets, Machine Learning (**ML**) and Programming Language (**PL**). In addition, **HEP-TH** [46] is a corpus of Physics papers with their citations. **Web** [45] is a Web page hyperlink network

Table 6.2: Classification accuracy (in percentage) on four subsets of test set with descending length. Best baselines are underlined. We show improvement of MCTM (G) over GATON and best baseline.

Category	Model	ML				
		Overall	0-25%	25%-50%	50%-75%	75%-100%
Models with plain text	ProdLDA	58.5±3.2	69.7±1.0	59.1±4.6	55.8±5.0	48.7±4.6
	WLDA	31.3±0.7	30.9±2.9	31.6±2.2	34.3±1.2	28.2±2.2
	GATON	60.3±2.0	69.3±2.7	62.8±2.5	57.4±1.8	50.8±6.6
	MCTM (G)	**67.1±2.1**	**73.7±3.6**	**68.7±4.3**	**66.4±3.0**	**60.1±1.9**
	MCTM (C)	66.9±1.4	73.6±1.5	68.3±2.0	65.2±1.9	58.9±1.8
	improvement	11.4%* 11.4%*	6.3%* 5.7%*	9.4%* 9.4%*	15.7%* 15.7%*	18.3%* 18.3%*
Models with word embeddings	ETM	50.6±2.2	60.4±3.3	52.9±2.5	48.8±2.1	39.4±3.0
	NSTM	45.2±2.6	53.6±1.9	42.4±7.5	43.0±2.3	41.1±4.7
	GATON+WE	63.8±1.5	72.4±2.3	67.0±3.2	60.5±2.2	54.8±2.3
	MCTM+WE (G)	**66.8±2.0**	**72.9±5.1**	**67.7±2.4**	65.3±3.8	**61.6±4.9**
	MCTM+WE (C)	66.0±1.4	70.9±1.9	67.1±3.1	**67.1±2.8**	58.3±2.7
	improvement	4.6%* 4.6%*	0.7% 0.7%	1.0% 1.0%	7.9%* 7.9%*	12.3%* 12.3%*
Models with document networks	RTM	64.2±2.3	72.9±3.6	71.0±1.7	61.5±4.0	50.6±3.4
	Adj-Enc	71.0±0.4	78.9±0.7	74.6±1.9	72.4±2.1	57.8±1.4
	LANTM	72.1±1.6	74.9±3.1	77.2±1.3	71.6±2.2	64.5±4.5
	GATON+DN	67.7±1.2	74.7±3.3	71.5±3.1	67.8±4.2	58.2±4.7
	meta-tail2vec	58.7±1.6	65.8±4.5	62.0±2.9	59.0±3.7	47.2±4.0
	MCTM+DN (G)	**83.3±1.7**	**86.2±2.9**	**82.7±2.1**	**81.9±2.6**	82.1±2.4
	MCTM+DN (C)	83.0±1.2	85.9±0.7	82.0±1.3	81.2±1.1	**82.8±3.9**
	improvement	22.9%* 15.4%*	15.3%* 9.2%*	15.6%* 7.1%*	20.9%* 13.1%*	41.1%* 27.3%*

Category	Model	PL				
		Overall	0-25%	25%-50%	50%-75%	75%-100%
Models with plain text	ProdLDA	45.0±2.1	51.4±6.0	48.7±3.6	41.6±2.3	40.1±3.9
	WLDA	33.2±1.8	37.4±1.5	40.0±5.3	28.1±3.9	23.8±1.8
	GATON	47.6±1.5	53.8±3.1	52.4±3.9	44.5±4.9	39.0±3.4
	MCTM (G)	**53.5±0.8**	59.8±3.5	**57.3±2.8**	**52.9±5.7**	**45.5±2.0**
	MCTM (C)	53.4±0.9	**60.6±3.1**	56.3±1.6	52.5±2.0	42.9±1.3
	improvement	12.3%* 12.3%*	11.2%* 11.2%*	9.3%* 9.3%*	18.8%* 18.8%*	16.8%* 13.6%*
Models with word embeddings	ETM	43.8±2.0	48.5±2.8	47.9±2.7	42.4±1.9	35.9±4.1
	NSTM	41.3±3.2	47.2±5.3	45.0±5.1	39.8±4.6	33.1±2.4
	GATON+WE	50.2±1.5	57.4±2.3	53.7±3.3	48.7±2.7	40.2±3.1
	MCTM+WE (G)	**52.7±1.7**	**60.7±6.1**	**53.9±3.1**	**51.8±1.6**	**43.8±2.6**
	MCTM+WE (C)	52.1±1.5	**60.7±2.7**	53.1±2.8	50.5±4.8	43.6±3.6
	improvement	5.0%* 5.0%*	5.7%* 5.7%*	0.3% 0.3%	6.3%* 16.3%*	9.0%* 9.0%*
Models with document networks	RTM	53.3±1.1	58.7±3.5	58.9±3.0	52.4±3.4	42.6±2.0
	Adj-Enc	60.4±1.1	63.6±1.9	63.8±1.5	62.6±2.1	52.1±3.7
	LANTM	60.8±0.9	66.5±3.1	61.4±1.5	62.4±1.7	52.4±3.7
	GATON+DN	58.5±2.0	65.4±2.0	62.2±3.1	61.0±1.7	44.3±3.8
	meta-tail2vec	44.9±3.0	51.9±2.6	48.7±3.1	46.6±4.1	31.4±6.8
	MCTM+DN (G)	**72.9±1.0**	**77.2±4.3**	**73.9±3.8**	**71.9±3.9**	68.1±2.8
	MCTM+DN (C)	71.9±0.7	73.7±3.3	72.9±3.0	71.0±2.1	**70.0±2.5**
	improvement	24.6%* 19.8%*	18.2%* 16.1%*	18.7%* 15.7%*	18.0%* 14.9%*	53.7%* 29.9%*

where each page is a news article, and the hyperlinks connect related articles. See Table 6.1 for details.

Baselines. Since our model has three variants, i.e., MCTM with plain texts, with auxiliary word embeddings, and with auxiliary document networks, we correspondingly compare to three categories of baselines. *i*) **Topic models with plain texts**, ProdLDA [76], WLDA [63], and GATON [100]. They model all documents uniformly without dealing with short texts. We compare to them and show the advantage of MCTM on improving short texts. *ii*) **Topic models with word embeddings**, ETM [19] and NSTM [118]. Since our model is built on top of GATON, we also compare to GATON with word embeddings, denoted as GATON+WE. By comparing to them, we verify the effectiveness of semantic complement meta-learning to further improve topic quality. *iii*) **Topic models with document networks**, RTM [10], Adjacent-Encoder [107], LANTM [90], and GATON+DN, which is the extension of GATON with document networks. For document network scenario, we include a graph embedding model, meta-tail2vec [53], which uses meta-learning to improve nodes with low degrees, but is not a topic model and ignores variable lengths of node attributes, i.e., texts.

We set $L = 2$ convolutional layers. $\lambda_{\text{gen}} = 2$ and $\lambda_{\text{reg}} = 0.05$. Number of negative samples $M = 5$ and number of semantic clusters $R = 5$. $\mathscr{L}$ is the median length of the corpus. Local and global learning rates are $\alpha_1 = 0.001$ and $\alpha_2 = 0.0005$. We use 300D Glove embeddings. We experiment with 5 independent runs, report mean and std.dev. All the experiments were done on Linux server with a Tesla K80 GPU with 11441MiB.

6.4.1 Quantitative Evaluation

Document Classification. Documents from the same category discuss related topics. As in LDA [6], we conduct classification to evaluate topic quality. We split 80% documents for training (10% are for validation). Labels are not involved during training. After convergence, we train a kNN classifier ($k = 5$) [4] with training documents and predict the labels of test documents. We

Table 6.3: Topic coherence NPMI (left, in percentage) and perplexity (right) at $K = 64$.

Category	Model	Topic Coherence NPMI				Perplexity			
		ML	PL	HEP-TH	Web	ML	PL	HEP-TH	Web
Models with plain text	ProdLDA	6.3±0.2	9.4±0.5	10.3±0.6	16.2±1.4	7.19±0.00	7.21±0.00	7.72±0.00	8.34±0.00
	WLDA	9.7±0.2	11.6±0.1	**13.7±0.4**	**23.9±0.8**	18.90±0.73	19.57±0.30	44.31±0.18	45.22±0.00
	GATON	9.9±0.9	8.4±1.5	8.9±1.5	4.8±1.1	9.64±0.27	9.17±0.10	8.79±0.57	8.52±0.12
	MCTM (G)	**10.0±1.4**	**12.1±1.2***	**13.7±1.7**	13.5±2.5	3.81±0.24	**3.60±0.51***	**3.98±0.29***	3.27±0.41
	MCTM (C)	9.9±2.0	12.0±1.8	13.2±2.1	16.1±1.1	**3.76±0.17***	3.63±0.20	4.12±0.30	**3.13±0.13***
Models with word embeddings	ETM	5.5±0.1	7.7±0.2	7.2±0.4	16.4±0.7	8.67±0.00	8.52±0.00	8.51±0.00	8.52±0.00
	NSTM	16.0±1.0	18.6±0.6	18.2±0.5	**27.9±0.6**	8.46±0.00	8.34±0.00	8.39±0.00	8.30±0.00
	GATON+WE	16.1±1.4	12.9±1.5	16.9±1.1	12.4±1.1	5.50±0.21	5.56±0.48	8.36±0.02	7.98±0.02
	MCTM+WE (G)	**17.6±1.2***	19.1±2.8	18.2±1.3	23.6±0.4	**4.43±0.17***	**4.23±0.56***	**3.36±0.10***	3.18±0.12
	MCTM+WE (C)	16.8±1.1	**20.3±1.5***	**18.7±0.9**	23.8±1.0	4.62±0.16	4.62±0.24	3.50±0.17	**3.04±0.12***
Models with document networks	RTM	7.3±0.2	8.9±0.5	6.6±0.3	**18.0±0.4**	8.07±0.01	7.93±0.01	8.04±0.00	8.96±0.13
	Adj-Enc	8.4±0.4	10.5±0.1	6.4±0.4	7.2±0.5	7.41±0.01	7.34±0.13	7.45±0.19	7.65±0.00
	LANTM	9.9±1.2	9.8±0.7	10.4±1.5	N.A.	8.63±0.00	8.48±0.00	8.50±0.00	N.A.
	GATON+DN	10.3±0.7	10.7±1.1	9.7±0.8	7.7±2.0	8.58±0.02	8.43±0.00	8.33±0.01	8.13±0.02
	MCTM+DN (G)	**11.9±2.0***	**11.1±2.0***	**10.6±1.7**	15.5±1.1	4.07±0.27	4.12±0.31	**3.99±0.40***	**3.21±0.25***
	MCTM+DN (C)	11.6±1.5	10.3±1.4	10.4±1.5	17.0±1.1	**3.41±0.36***	**3.63±0.46***	4.13±0.23	3.75±0.43

set 64 topics. We compare our models within each category of baselines and report classification accuracy at Table 6.2. We split the test set into four subsets with descending document length and report the result of both overall test and each subset. 0-25% at Table 6.2 means the subset with the longest 25% test documents. MCTM (G) and MCTM (C) denote our model with GNN and Clustering function, respectively. We use "*" to represent statistically significant improvement with paired t-test at 0.05 significance level.

Our models significantly outperform baselines within each category. We outperform GATON, the best baseline in the plain text and word embedding category, since textual semantic complement improves short texts, and the overall test set is also improved. MCTM (G) performs slightly better than MCTM (C), potentially because GNN recognizes importance of words with attention, while the Clustering function takes simple average. To show our models indeed improve short text quality, we present the improvement of MCTM (G) over both GATON and the best baseline. Our performance generally improves more as the length decreases, which verifies the advantage of semantic complement.

Topic Coherence. Each row of topic-word distribution $\mathbf{Z}_{\mathcal{V}} \in \mathbb{R}^{K \times |\mathcal{V}|}$ is the distribution of one topic over the vocabulary, and the key words of this topic correspond to the highest values on this row. As in ProdLDA [76], we evaluate the coherence of key words by Google Web 1T

Table 6.4: Topic interpretability.

Topic	Key words of MCTM (G)
1	variance, probability, generalize, covariance, approximation
2	non-genetic, rnn, stimulus-response, epistasis, mismatch

Topic	Key words of MCTM (C)
1	scalability, multiprocessor, obviate, compute, algorithm
2	sphere, tangent, three-dimensional, vector, geometrical

5-gram Version 1 [22], with NPMI as metric. Table 6.3 (left) summarizes the results. Topic-word distribution $\mathbf{Z}_{\mathcal{V}}$ is model parameter and is separate from document length, thus we can not report results of different lengths. LANTM cannot run on large dataset Web. Meta-tail2vec is not a topic model, thus is excluded. Overall, our models outperform baselines on ML and PL and are competitive with the best baseline on HEP-TH and Web. This indicates that our models at least do not hurt topic coherence, but can significantly improve other tasks, e.g., classification. Compared to GATON, our models significantly improve it, verifying the advantage of semantic complement. To understand what topics our models capture, we randomly present two topics with top-5 key words at Table 6.4. MCTM (G) captures *statistics* and *computational genetics*, while MCTM (C) reveals *scalability* and *geometric learning*.

Perplexity. Topic model should generalize to unseen documents. Following [6], we evaluate perplexity. Since perplexity is exponential and varies much w.r.t. its power, we report its power, $-\frac{\log \Pr(\mathcal{D}_{\text{test}})}{\sum_{d \in \mathcal{D}_{\text{test}}} l_d}$ (lower is better). Table 6.3 (right) reveals that our models generate high likelihood to unseen documents, which we attribute to semantic complement meta-learning module.

Link Prediction. A good model should infer similar topics for potentially linked documents. Since we model document network as auxiliary data, we follow RTM [10] and predict links. As in [107], the probability of a link is $p(e_{d,d'}) \propto \exp(-||\mathbf{z}_d^* - \mathbf{z}_{d'}^*||_2^2)$. We predict the links within test documents and compare with the ground-truth links with AUC as metric. Since only the third category (network models) incorporate links, we mainly compare the MCTM+DN version to these network baselines. Table 6.5 indicates that our models predict links more accurately than

Table 6.5: Link prediction (in percentage) on overall test set and the shorter half of the test set. Best baselines are underlined. We show the improvement of MCTM (G) over GATON and best baseline.

Model	ML		PL		HEP-TH		Web	
	Overall	Short	Overall	Short	Overall	Short	Overall	Short
RTM	71.4±0.9	67.0±0.6	68.2±0.4	62.2±0.3	69.7±0.8	65.1±0.6	69.9±0.1	75.3±0.1
Adj-Enc	88.1±0.2	86.2±0.3	79.6±0.3	73.8±0.2	88.9±0.1	88.4±0.1	82.7±0.1	78.5±0.2
LANTM	76.5±1.2	76.1±1.6	73.9±0.9	70.5±1.1	86.6±0.0	85.2±0.0	N.A.	N.A.
GATON+DN	74.2±0.4	71.6±0.9	71.6±0.8	65.8±0.6	90.1±0.2	89.1±1.2	74.3±0.2	71.7±0.3
meta-tail2vec	69.7±2.3	66.5±1.5	68.7±1.7	64.3±1.5	N.A.	N.A.	N.A.	N.A.
MCTM+DN (G)	**94.0±0.5**	91.9±1.4	**91.5±0.1**	**90.6±0.8**	**93.9±0.2**	**93.9±0.1**	**83.4±0.1**	**80.5±0.1**
MCTM+DN (C)	93.4±0.5	**92.7±0.7**	91.2±0.8	90.2±0.9	92.5±0.3	92.4±0.4	80.6±0.2	77.5±0.2
improvement	26.6%* 6.7%*	28.4%* 6.6%*	27.9%* 15.0%*	37.7%* 22.8%*	4.2%* 4.2%*	5.4%* 5.4%*	12.2%* 0.8%*	12.3%* 2.5%*

Table 6.6: Effect of semantic complement and meta-learning on document classification on ML.

Model	Effect of Semantic Complement				Effect of Meta-Learning				Effect of both	
	Overall test set		Short subset		Overall test set		Short subset		Test	Short
	with	without	with	without	with	without	with	without	without	without
MCTM (G)	**67.1±2.1**	58.2±2.4	**60.1±1.9**	50.8±3.8	**67.1±2.1**	65.9±1.3	**60.1±1.9**	57.0±3.5	52.1±4.1	44.5±5.5
(decline)		(13.3%*)		(15.5%*)		(1.8%)		(5.2%*)	(22.4%*)	(26.0%*)
MCTM (C)	**66.9±1.4**	56.7±3.1	**58.9±1.8**	49.4±2.9	**66.9±1.4**	65.6±1.0	**58.9±1.8**	56.1±2.4	52.5±2.4	46.9±4.8
(decline)		(15.2%*)		(16.1%*)		(1.9%)		(4.8%*)	(21.5%*)	(20.4%*)

baselines. The comparison to network baselines demonstrates the effectiveness of both textual and structural semantic complement.

6.4.2 Model Analysis

To better understand our model, we conduct model analysis here.

Effect of Semantic Complement. To see if semantic complement helps short texts, we remove it from the complete model and present classification accuracy at Table 6.6 (left). We show the plain text results, and put the auxiliary data version in supplementary. Models do better with semantic complement than without. The accuracy on short subset declines more than the overall test set, which reveals that semantic complement improves short texts, and removing it hurts short documents more.

Effect of Meta-Learning. To analyze if meta-learning benefits the optimization with a few observed words, we replace it with the commonly used stochastic gradient descend. Table 6.6 (middle) shows that result drops more on short subset than on the overall test set, which verifies that meta-learning is good at optimization with only a few observed words and improves short

Table 6.7: Effect of scaling-and-shifting and clustering.

Model	Scaling and Shifting		Effect of Clustering	
	with	without	with	without
MCTM	**67.1±2.1**	65.7±2.9	**66.9±1.4***	62.7±1.6
MCTM+WE	**66.8±2.0**	65.9±1.0	**66.0±1.4***	62.4±1.9
MCTM+DN	**83.3±1.7**	82.4±1.4	**83.0±1.2***	81.0±1.3

text modeling. We further remove both semantic complement and meta-learning, and report the result at Table 6.6 (right), which presents the worst accuracy. This observation further verifies that both components are important.

Effect of Scaling and Shifting. The GNN version of our model uses scaling-and-shifting method to recover distinct topics. To test its usefulness, we disregard it and summarize classification accuracy on ML at Table 6.7 (left). Removing scaling and shifting leads to worse performance, since we can not personalize shared parameters to each document to recover its distinct missing topics for complement.

Effect of Clustering. We set the number of clusters R to 1, all documents share the same parameters for missing semantics prediction with no clustering. Table 6.7(right) concludes that clustering is helpful to share common semantics for related documents and distinguish documents of different clusters.

6.5 Discussion

We improve short text topic modeling with semantic complement meta-learning. We complement the semantics for short documents by contrastive learning and design two alternatives for missing semantics prediction. Meta-learning helps to optimize and predict the missing semantics. Experiments on document classification, topic coherence, perplexity, and link prediction verify the effectiveness of our model. One limitation is to assume a variable-length corpus with both long and short documents for semantic transfer. We also assume the content is truthful. If the corpus is infiltrated by fake news, those may appear in some topics.

Chapter 7

Topic Modeling on Document Networks with Dirichlet Optimal Transport Barycenter

7.1 Introduction

While text documents are primarily expressed by words, in many cases they are also interconnected in a network structure. For example, academic papers constitute a citation network, Web pages present a hyperlink network, user profiles are connected in a social network. Graph Neural Networks (GNNs) [40] are powerful tools to derive effective low-dimensional embeddings for such networked documents, which could fulfill downstream tasks, such as document classification and link prediction. However, when dealing with text documents, we usually model a latent topic structure [6] where each document is represented by a low-dimensional topic distribution, and each topic is characterized by a group of understandable key words. Most previous GNNs ignore such topic structure, resulting in *uninterpretable* embeddings.

Topic modeling provides an appealing method to uncover latent, semantically interpretable topics that occur in a text corpus. However, many existing topic modeling works, e.g., LDA [6],

deal with the plain text within each document only, without considering *network connectivity* across documents. Intuitively, two connected documents are likely to share similar topics, e.g., two hyperlinked news articles tend to report similar events. Modeling document network structure in addition to the textual content could discover meaningful semantics and improve topic quality.

In this chapter, we investigate the design of Optimal Transport Barycenter for short text modeling on document networks. Different from Chapter 6, which uses meta-learning to transfer the semantic knowledge within the corpus, without auxiliary information needed, this chapter leverages external knowledge, pre-trained word embeddings, to approach short text topic modeling.

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

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

7.2 Background

This work is built on top of Optimal Transport Barycenter. Optimal Transport (OT) has been introduced at Sec. 4.2. Here we introduce OT Barycenter, as well as rejection sampling to be used.

Definition 7.2.1 (Optimal Transport Barycenter)**.** *Given a set of distributions* $\{\boldsymbol{q}_i\}_{i=1}^Q$*, and weights*

of measures $\{a_i\}_{i=1}^{Q}$ *where* $a_i > 0$ *and* $\sum_{i=1}^{Q} a_i = 1$*, OT barycenter is defined as*

$$\arg\min_{\boldsymbol{p}} \sum_{i=1}^{Q} a_i d_{\boldsymbol{C}}(\boldsymbol{p}, \boldsymbol{q}_i). \tag{7.1}$$

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

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

Definition 7.2.2 (Rejection Sampling)**.** *Rejection sampling allows to sample data points from a relatively complex distribution, e.g., Dirichlet distribution, to enable reparameterization. Specifically, since directly sampling from a target distribution* $p(\boldsymbol{z})$ *is difficult, we instead seek another proposal function* $f(\boldsymbol{z})$*, which may not be a probability distribution, such that* $f(\boldsymbol{z}) \geq p(\boldsymbol{z})$ *and sampling from* $f(\boldsymbol{z})$ *is possible. Rejection sampling contains three steps:*

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

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

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

7.3 Model Architecture and Analysis

Our work is built on Variational Graph Auto-Encoder [40]. We first use a graph convolutional encoder to project documents into low-dimensional topic space. For a document i,

$$\tilde{\mathbf{h}}_i^{(l)} = \mathbf{W}^{(l)}\mathbf{h}_i^{(l-1)}. \tag{7.2}$$

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

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

$$a_{ij} = \text{softmax}\Big(\text{sigmoid}(\mathbf{b}^{(l)\top}[\tilde{\mathbf{h}}_i^{(l)}||\tilde{\mathbf{h}}_j^{(l)}])\Big) \quad \text{where} \quad j \in \mathcal{N}(i). \tag{7.3}$$

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

$$\mathbf{h}_i^{(l)} = f_{\text{act}}\Big(\frac{1}{2}(\tilde{\mathbf{h}}_i^{(l)} + \sum_{j\in\mathcal{N}(i)} a_{ij}\tilde{\mathbf{h}}_j^{(l)})\Big). \tag{7.4}$$

$f_{\text{act}}(\cdot)$ is activation. We set it to identity function $f_{\text{act}}(x) = x$ only for the final convolutional layer and $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ for other layers. We obtain $\mathbf{h}_i^{(l)}$ as the output from the l-th layer, containing both text content and i's neighborhood. We repeat convolutional process at Eq. 7.2–7.4 for maximum L layers, and obtain $\mathbf{h}_i = \mathbf{h}_i^{(L)} \in \mathbb{R}^K$ as the output from the encoder. K is the number of topics.

We aim to use $\mathbf{h}_i$ as concentration parameter of Dirichlet distribution to draw document i's topic distribution, i.e., $\mathbf{z}_i \sim \text{Dir}(\mathbf{h}_i)$. However, Dirichlet's concentration parameter should be positive, while $\mathbf{h}_i$ from Eq. 7.4 is obtained by identity function with both positive and negative

values. To solve this problem, we have

$$\boldsymbol{\alpha}_i = \max(10^{-12}, \text{softplus}(\mathbf{h}_i)). \tag{7.5}$$

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

7.3.1 Dirichlet Reparameterization

Since Dirichlet prior in LDA [6] has shown promise in improving topic quality, we are motivated to use Dirichlet as a prior for topic distribution to enhance topic quality, $\mathbf{z}_i \sim \text{Dir}(\boldsymbol{\alpha}_i)$. However, Dirichlet is not location scale family, thus it is difficult to directly sample topic distributions from it, which hinders reparameterization. To solve this problem, we use rejection sampling [4].

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

$$\mathbf{z}_i = \left[\frac{z_{i,k}}{\sum_{k'=1}^{K} z_{i,k'}}, ..., \frac{z_{i,K}}{\sum_{k'=1}^{K} z_{i,k'}} \right] \sim \text{Dir}(\boldsymbol{\alpha}_i). \tag{7.6}$$

As long as we can sample topics from Gamma $\Gamma(\alpha_{i,k})$, we can approximate Dirichlet by the normalization at Eq. 7.6. Now the problem is how to sample topics from Gamma, i.e., $z_{i,k} \sim \Gamma(\alpha_{i,k})$.

Fortunately, there exits a proposal function for Gamma distribution $\Gamma(\alpha_{i,k})$ below [8], such that $f(\alpha_{i,k}) \geq \Gamma(\alpha_{i,k})$.

$$z_{i,k} = f(\alpha_{i,k}) = (\alpha_{i,k} - \frac{1}{3})(1 + \frac{\epsilon}{\sqrt{9\alpha_{i,k} - 3}})^3, \quad \epsilon \sim \mathcal{N}(0, 1). \tag{7.7}$$

Sampling from proposal function $f(\alpha_{i,k})$ is possible by first sampling a variable ϵ from a Gaus-

sian $\mathcal{N}(0,1)$, then putting into Eq. 7.7 to obtain the sampled topics $z_{i,k}$. However, sampled topics $z_{i,k}$ from this proposal function do not strictly obey the target Gamma distribution, since some samples should be rejected. Lower the concentration parameter $\alpha_{i,k}$, lower the acceptance rate [8]. If the acceptance rate is overly low, we need multiple repetitions of rejection sampling to obtain one valid sample. To this end, we aim to increase the acceptance rate by increasing concentration parameter $\alpha_{i,k}$. Fortunately, [62] suggests the following solution: since we seek to sample topics $z_{i,k}$ from Gamma distribution by $z_{i,k} \sim \Gamma(\alpha_{i,k})$, we can equivalently rewrite this sampling by

$$z_{i,k} = \bar{z}_{i,k} \prod_{c=1}^{C} u_c^{\frac{1}{\alpha_{i,k}+c-1}} \quad \text{where} \quad \bar{z}_{i,k} \sim \Gamma(\alpha_{i,k} + C). \tag{7.8}$$

Positive integer C is a hyperparameter to boost concentration parameter $\alpha_{i,k}$, so that sampling process $\bar{z}_{i,k} \sim \Gamma(\alpha_{i,k} + C)$ has a higher concentration parameter $\alpha_{i,k} + C$ now, which leads to a higher acceptance rate by rejection sampling at Eq. 7.7. As in [62], $C = 10$ provides an acceptance rate higher than 0.99, thus we set $C = 10$ and accept all samples for simplicity. After obtaining $\bar{z}_{i,k}$ as a valid sample at Eq. 7.7, we use Eq. 7.8 to calculate $z_{i,k}$. Above, we use one topic $z_{i,k}$ for illustration. We repeat this process for K (the number of topics) times, and obtain K valid sampled topics $\{z_{i,k}\}_{k=1}^{K}$ from Gamma. Finally, we use Eq. 7.6 to normalize them and obtain K-dimensional topic distribution $\mathbf{z}_i$. To summarize, $\mathbf{z}_i \sim \text{Dir}(\boldsymbol{\alpha}_i) = q(\mathbf{z}_i)$. Variational posterior $q(\mathbf{z}_i)$ contains both graph convolutional encoder and Dirichlet reparameterization.

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

$$\begin{aligned} \text{KL}[q(\mathbf{z}_i)||p(\mathbf{z})] &= \log\Gamma(\sum_{k=1}^{K}\alpha_{i,k}) - \log\Gamma(\sum_{k=1}^{K}\alpha_k^0) + \sum_{k=1}^{K}\log\Gamma(\alpha_k^0) \\ &- \sum_{k=1}^{K}\log\Gamma(\alpha_{i,k}) + \sum_{k=1}^{K}(\alpha_{i,k} - \alpha_k^0)(\Psi(\alpha_{i,k}) - \Psi(\sum_{k=1}^{K}\alpha_{i,k})) \end{aligned} \tag{7.9}$$

where $\Psi(\cdot)$ is digamma function [8].

7.3.2 Barycentric Decoding (DBN)

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

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

$$\min \sum_{j \in \mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j). \tag{7.10}$$

a_{ij} is defined at Eq. 7.3. By moving topic $\mathbf{z}_i$ to i's neighbors $\mathcal{N}(i)$, $\mathbf{z}_i$ becomes the barycenter of $\mathcal{N}(i)$. Network structure is captured.

We now define the cost matrix at Eq. 7.10. Each element c_{kw} in $\mathbf{C} \in \mathbb{R}^{K \times |\mathcal{V}|}$ specifies the dissimilarity between topic k and word w. We set $c_{kw} = 1 - \cos(\mathbf{t}_k, \mathbf{e}_w)$ where $\mathbf{t}_k$ is the randomly initialized topic embedding, and $\mathbf{e}_w$ is pre-trained word embedding [61, 68]. External knowledge is naturally incorporated by cost matrix, and helps alleviate word sparsity problem of short documents.

Decoding. Eq. 7.10 pushes topic distribution $\mathbf{z}_i$ towards the word distribution of i's neighbors, which is similar to a decoder. We explicitly design another decoder $\hat{\mathbf{d}}_i = \phi(\mathbf{z}_i) = \text{softmax}((2-\mathbf{C})^\top \mathbf{z}_i)$ to generate the content of *neighboring documents*. $2 - \mathbf{C}$ is decoding parameter, since cost matrix $\mathbf{C}$ captures topic-word distribution. $\mathbf{C}$ is defined by $c_{kw} = 1 - \cos(\mathbf{t}_k, \mathbf{e}_w)$, where cosine similarity has range $[-1, 1]$, thus c_{kw} has range $[0, 2]$. As in [6, 60], decoding parameter is a positive matrix where each element is the probability of a word belonging to a certain topic. We define $2 - \mathbf{C}$ to make our decoding parameter positive.

The log-likelihood is $\sum_{j \in \mathcal{N}(i)} a_{ij} \mathbf{d}_j^\top \log \hat{\mathbf{d}}_i = \sum_{j \in \mathcal{N}(i)} a_{ij} l(\mathbf{z}_i, \mathbf{d}_j)$. Weights a_{ij} are Eq. 7.3.

We generate content of neighbors in a 1-to-N process. Loss function is

$$\begin{aligned}\mathcal{J}_{\text{DBN}} = \sum_{j \in \mathcal{N}(i)} \mathbb{E}_{\mathbf{z}_i \sim q(\mathbf{z}_i)}[a_{ij}(-l(\mathbf{z}_i, \mathbf{d}_j) + \lambda_{\text{OT}} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j))] \\ + \lambda_{\text{KL}} \text{KL}[q(\mathbf{z}_i)||p(\mathbf{z})].\end{aligned} \tag{7.11}$$

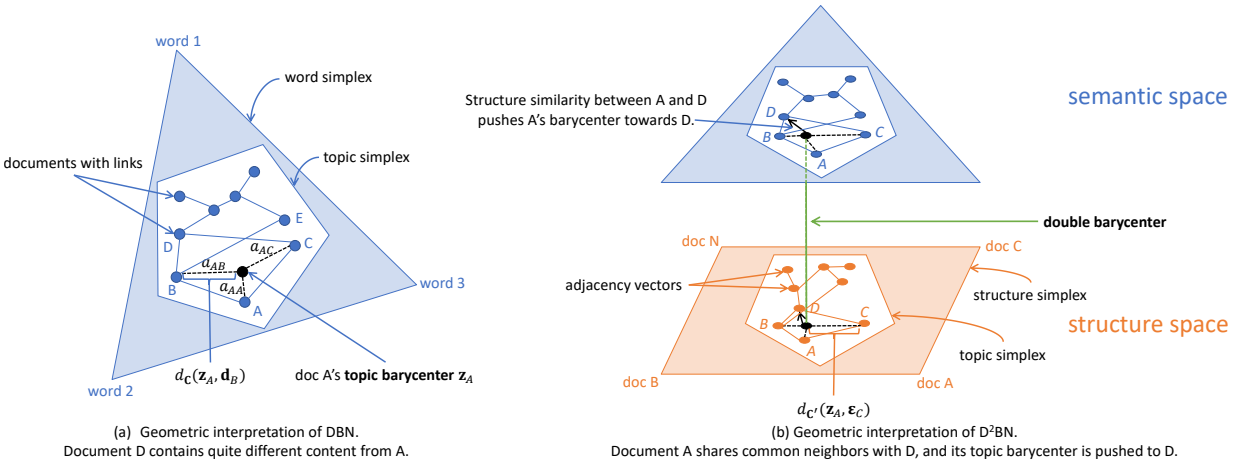
$q(\mathbf{z}_i)$ contains both graph convolutional encoder and Dirichlet reparameterization. Hyperparameter λ_{OT} balances log-likelihood and optimal transport barycenter, and λ_{KL} controls KL divergence.

Compared to previous topic models with OT, e.g., NSTM [118], we point out two main extensions. First, NSTM models each document individually and does not have document network, while we design a graph convolutional encoder to model both text and network connectivity. Second, motivated by the success of Dirichlet prior in LDA [6], we design a Dirichlet optimal transport prior with rejection sampling to improve topic quality. In contrast, NSTM does not impose any prior, likely suffering over-fitting and worse topic interpretability.

Compared to VGAE, whose embeddings do not enjoy semantic interpretability, the decoding term of our model at Eq. 7.11 contains one more component, optimal transport barycenter, which incorporates pre-trained word embeddings to define cost matrix $\mathbf{C}$ for topic modeling. Decoding parameter $2-\mathbf{C} \in \mathbb{R}^{K \times |\mathcal{V}|}$ corresponds to topic-word distribution. Each row is the distribution of a topic over the vocabulary, and the key words of that correspond to the highest values on that row. Thus the learned topic distributions are semantically interpretable by topic-word distribution.

For better understanding the connection between 1-to-N and 1-to-1 content generation, we have the below theorem.

Theorem 2. *Given $\boldsymbol{C}$, $|\mathcal{V}| \geq 8$, and a_{ij}, let $\boldsymbol{z}_j = \arg\min_{\boldsymbol{z}_j} d_{\boldsymbol{C}}(\boldsymbol{z}_j, \boldsymbol{d}_j)$ be the best solution of its individual optimal transport without barycenter, $\boldsymbol{z}_i = \arg\min_{\boldsymbol{z}_i} \sum_{j \in \mathcal{N}(i)} a_{ij} d_{\boldsymbol{C}}(\boldsymbol{z}_i, \boldsymbol{d}_j)$ be the best*



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

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

Figure 7.1: Geometric interpretation of DBN and D^{2}BN.

solution of optimal transport barycenter Eq. 7.10, we have

$$
\begin{aligned}
-\sum_{j\in\mathcal{N}(i)} a_{ij} l(\boldsymbol{z}_i, \boldsymbol{d}_j) &\geq \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\boldsymbol{C}}(\boldsymbol{z}_i, \boldsymbol{d}_j) \\
&\geq \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\boldsymbol{C}}(\boldsymbol{z}_j, \boldsymbol{d}_j).
\end{aligned}
\tag{7.12}
$$

The first inequality is similar to [118], OT barycentric distance $\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j)$ is the lower bound of the negative log-likelihood $-\sum_{j\in\mathcal{N}(i)} a_{ij} l(\mathbf{z}_i, \mathbf{d}_j)$. Thus, OT barycentric distance at Eq. 7.11 is a regularizer to avoid negative log-likelihood becoming overly low and resulting in over-fitting. Here, the constraint of vocabulary size, $|\mathcal{V}| > 8$, is required at one step in the proof to make sure the inequality is satisfied (Eq. 7.20). Almost all the real-world corpora have hundreds or thousands of words in vocabulary, thus $|\mathcal{V}| > 8$ is not a strict constraint, i.e., the first inequality is satisfied most of the time. The second inequality reveals that 1-to-N content generation by optimizing barycenter $\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j)$ is the upper bound of traditional 1-to-1 content generation without network structure, and provides a tighter regularizer. See Appendix for complete proofs.

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

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

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

```
Initialize all parameters.
while not converged do
    Sample a batch of documents {d_b}_{b=1}^B.
    Encode {d_b}_{b=1}^B and neighbors {N(b)}_{b=1}^B by q(·).
    Generate neighboring content by φ(·), and neighboring adjacency vector by σ(·) for D²BN.
    Obtain optimal transport plan T*_ij for DBN, and T'*_ij for D²BN, by solving OT d_C(z_i, d_j)
    and d_C'(z_i, ε_j) using Sinkhorn Iteration at Algo. 3, for each document i and each of its
    neighbors j ∈ N(i).
    Loss function Eq. 7.11 for DBN, Eq. 7.15 for D²BN.
    Update parameters by Adam optimizer.
end while
```

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

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

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

$$\sum_{j\in\mathcal{N}(i)} \mathbb{E}_{\mathbf{z}_i\sim q(\mathbf{z}_i)}[a_{ij}(-l(\mathbf{z}_i, \boldsymbol{\varepsilon}_j) + \lambda_{\text{OT}} d_{\mathbf{C}'}(\mathbf{z}_i, \boldsymbol{\varepsilon}_j))]. \tag{7.13}$$

Algorithm 3 Sinkhorn Iteration of DBN and D^{2}BN

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

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

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

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

We first define cost matrix $\mathbf{C}' \in \mathbb{R}^{K \times N}$ of $d_{\mathbf{C}'}(\mathbf{z}_i, \boldsymbol{\varepsilon}_j)$ at Eq. 7.13. Each element c'_{ki} preserves structural information between topic k and document i. $N = |\mathcal{D}|$ is the total number of documents. To keep consistent with $\mathbf{C}$ above, we similarly define $c'_{ki} = 1 - \cos(\mathbf{t}_k, \mathbf{s}_i)$. $\mathbf{t}_k$ is the same topic embedding as above, and $\mathbf{s}_i$ is the structure embedding of document i.

$$\mathbf{s}_i = f(\mathbf{W}_s \mathbf{x}_i + \mathbf{b}_s). \tag{7.14}$$

Consistently with pre-trained word embeddings, $\mathbf{x}_i$ is pre-trained structure embedding. Different from $\mathbf{x}_i$, $\mathbf{s}_i$ is the projected structure embedding and is derived by one-layer neural network. We feed $\mathbf{x}_i$ by pre-trained structure embeddings, e.g., DeepWalk [69]. $\mathbf{W}_s$ and $\mathbf{b}_s$ are parameters that project $\mathbf{x}_i$ to the same topic embedding space. The structure decoder is $\hat{\boldsymbol{\varepsilon}}_i = \sigma(\mathbf{z}_i) = \text{softmax}((2 - \mathbf{C}')^\top \mathbf{z}_i)$, log-likelihood at Eq. 7.13 is $l(\tilde{\mathbf{z}}_i, \boldsymbol{\varepsilon}_j) = \boldsymbol{\varepsilon}_j^\top \log \hat{\boldsymbol{\varepsilon}}_i$.

Finally, integrating both content and structure modeling, we have loss function for double

barycentric topic modeling D^{2}BN.

$$\mathcal{J}_{\text{D}^2\text{BN}} = \sum_{j\in\mathcal{N}(i)} \mathbb{E}_{\mathbf{z}_i\sim q(\mathbf{z}_i)}\Big[a_{ij}\Big(- l(\mathbf{z}_i, \mathbf{d}_j) + \lambda_{\text{OT}} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \\ +\lambda_s(-l(\mathbf{z}_i, \boldsymbol{\varepsilon}_j) + \lambda_{\text{OT}} d_{\mathbf{C}'}(\mathbf{z}_i, \boldsymbol{\varepsilon}_j))\Big)\Big] + \lambda_{\text{KL}}\text{KL}[q(\mathbf{z}_i)||p(\mathbf{z})]. \tag{7.15}$$

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

7.3.4 Optimization and Analysis

Geometric interpretation. For DBN at Fig. 7.1(a), the black dot is document A's topic distribution, which is also the topic barycenter of A and its neighbors B and C. We minimize weighted OT distance between $\mathbf{z}_A$ and not only $\mathbf{d}_A$, but also $\mathbf{d}_B$ and $\mathbf{d}_C$. For D^{2}BN at Fig. 7.1(b), structure space contains adjacency vectors, which regularize topic distributions. Although D contains different content from A in semantic space, they share common neighbors in structure space. A's topic distribution is also pushed to D. Such a double barycenter enjoys both semantic and structure information.

Optimization. We summarize the learning process in Algo. 2. Here we apply minibatch optimization. Line 6 calculates optimal transport plan $\mathbf{T}^*$ and $\mathbf{T}'^*$ by Sinkhorn iteration [14] at Algo. 3. After training convergence, we infer the topic distribution of a previously unseen document $\mathbf{d}'$ simply by the encoder $\mathbf{z}' \sim q(\mathbf{z}')$.

Complexity. To better understand our models, we provide computational complexity here. Encoding has $\mathcal{O}(\deg^L_{\max}(WK + \deg_{\max} K))$. $\deg_{\max}$ is the maximum number of neighbors, and W is the dimension of word embeddings. Dirichlet reparameterization and KL divergence has $\mathcal{O}(CK + \alpha^0 K + K^2)$. Here, $\boldsymbol{\alpha}^0 = [\alpha^0_1, ..., \alpha^0_K]$ is the concentration parameter of Dirichlet prior where $\alpha^0_1 = ... = \alpha^0_K = \alpha^0$. Decoder is $\mathcal{O}(K|\mathcal{V}|)$ for DBN and $\mathcal{O}(K(|\mathcal{V}| + N))$ for D^{2}BN. OT optimization is $\mathcal{O}(WK|\mathcal{V}|)$ for DBN and $\mathcal{O}(WK(|\mathcal{V}| + N))$ for D^{2}BN. Putting all components

Table 7.1: Dataset statistics.

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

together and removing trivial terms, we have $\mathcal{O}(\deg_{\max}^{L}(WK + \deg_{\max} K) + K^2 + WK|\mathcal{V}|)$ for DBN and $\mathcal{O}(\deg_{\max}^{L}(WK + \deg_{\max} K) + K^2 + WK(|\mathcal{V}| + N))$ for D^{2}BN.

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

7.4 Experiments

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

Datasets. Cora [58] is a corpus of academic papers with abstract as content and citations as links. We create three independent datasets, Data Structure (**DS**), Machine Learning (**ML**), and Programming Language (**PL**). **Aminer** [78] is another citation network with titles as the only content. We further create a **Web** page hyperlink network [45]. Each page is a news article. Hyperlinks point to relevant pages. Table 7.1 shows statistics.

Baselines. We consider three categories of baselines models.

1. **Topic models without network structure**, ProdLDA [76], Dirichlet VAE [8], ETM [19], WLDA [63], NSTM [118], and GTM [119]. ProdLDA and DVAE use Dirichlet as prior. ETM is a neural model with pre-trained word embeddings, WLDA applies Wasserstein

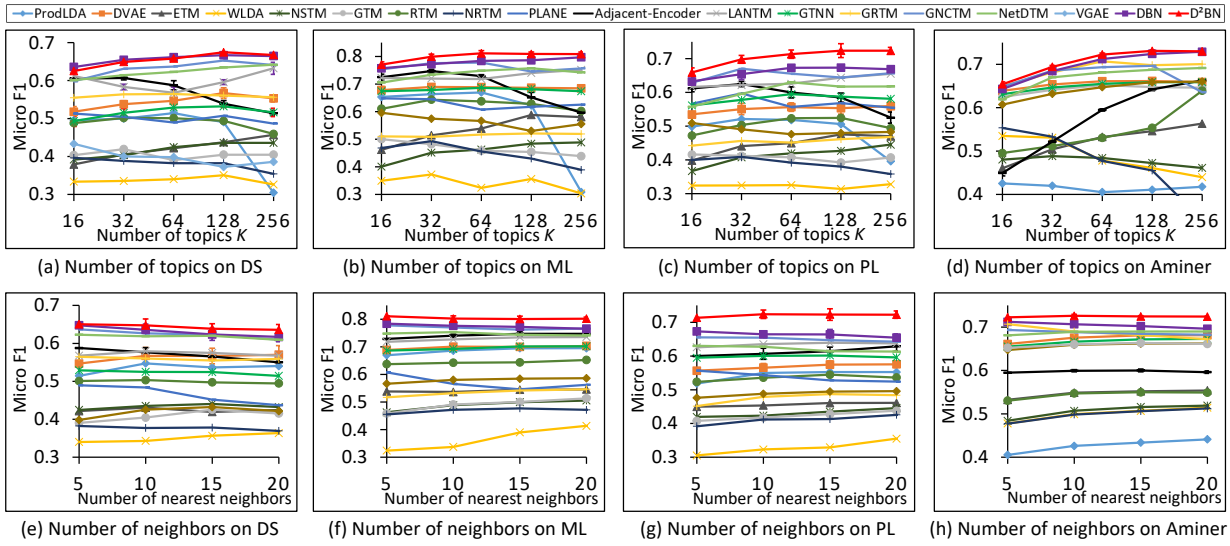


Figure 7.2: Document classification with Micro F1 score when varying number of topics (a-d) and number of nearest neighbors κ for κNN (e-h).

distance, and NSTM uses optimal transport. GTM is designed with graph neural networks. They do not incorporate network structure. By comparison, we show the advantage of jointly modeling text and network connectivity.

2. **Topic models for document networks**, RTM [10], NRTM [2], Adjacent-Encoder [107], LANTM [90], GTNN [94], GRTM [93], GNCTM [96], and NetDTM [113]. NRTM and Adjacent-Encoder are built on Auto-Encoders. LANTM, GTNN, GRTM, and GNCTM are designed with graph neural networks. They consider both text and links, but no one models pre-trained word embeddings. The comparison to them validates the effect of pre-trained word embeddings. LANTM extends VGAE with Gaussian prior. The comparison to LANTM shows the utility of Dirichlet prior. NetDTM is designed with pre-trained word embeddings, but not OT barycenter.

3. **Graph embedding.** There are models that learn node embeddings on attributed graphs in an unsupervised way. Strictly speaking, they are not topic models, nor baselines. For completeness, we still compare to VGAE [40].

Table 7.2: Document classification with Micro F1 (left) and Macro F1 (right) at $K = 64$. Results are in percentage. LANTM cannot run on Aminer even on a machine with 256GB. Web dataset does not have ground-truth labels, thus can not evaluate document classification.

Category	Model	Micro F1 score				Macro F1 score			
		DS	ML	PL	Aminer	DS	ML	PL	Aminer
Topic models w/o network structure	ProdLDA	51.4±1.1	67.0±0.6	51.8±0.6	40.5±0.0	40.1±4.3	67.1±0.7	48.5±1.2	14.6±0.9
	DVAE	54.7±2.2	68.9±0.6	55.7±1.3	66.1±0.5	45.8±1.2	65.5±1.4	49.3±1.5	44.6±1.6
	ETM	42.2±2.4	53.9±1.8	45.0±2.1	53.2±0.7	31.1±3.4	50.4±1.7	40.3±1.3	25.7±0.8
	WLDA	34.0±3.6	32.4±1.1	30.5±3.1	47.8±2.1	24.7±3.0	30.5±1.0	24.3±3.8	18.6±1.9
	NSTM	42.5±1.8	46.3±1.8	42.0±1.6	48.4±0.3	34.9±3.3	42.9±1.8	34.6±0.7	20.0±0.6
	GTM	39.1±1.6	45.9±1.4	40.8±1.6	65.2±0.2	29.6±1.3	42.1±1.7	34.4±2.0	43.3±0.7
Topic models for document networks (LANTM cannot run on large dataset Aminer)	RTM	50.1±2.3	63.7±0.9	52.3±0.7	53.0±0.7	41.3±2.2	58.6±1.6	44.4±0.7	25.1±0.7
	NRTM	38.2±1.2	45.5±1.2	39.2±1.5	47.7±3.9	32.3±1.6	41.7±1.4	34.1±0.9	13.5±3.2
	Adjacent-Encoder	58.8±1.2	72.8±0.6	60.0±1.7	59.5±0.2	54.6±1.5	72.8±0.8	55.3±1.6	46.9±0.5
	LANTM	56.8±2.4	71.8±1.0	62.6±1.3	N.A.	54.7±0.8	70.0±1.3	55.6±1.9	N.A.
	GTNN	52.9±1.4	68.6±1.1	59.5±2.3	65.5±0.4	42.8±3.3	67.8±1.1	59.4±1.5	41.8±0.5
	GRTM	56.5±1.9	51.7±2.3	45.2±1.0	70.7±0.2	50.2±2.0	48.3±1.9	37.5±2.1	48.8±0.5
	GNCTM	63.7±1.4	77.8±1.1	65.5±3.8	69.3±1.2	59.6±2.0	75.9±2.8	60.1±4.0	47.6±2.6
	NetDTM	62.3±1.0	74.8±1.0	63.0±1.1	68.1±0.1	58.0±1.4	73.1±1.1	57.2±1.2	46.6±0.7
Graph embedding	VGAE	39.8±2.0	56.6±1.7	47.6±3.4	64.7±0.5	40.3±3.5	61.5±2.3	50.0±1.5	45.0±1.4
Our proposed models	DBN	**66.2±1.4**	78.4±0.8	67.3±0.5	71.2±0.4	**62.7±1.6**	77.3±0.6	61.2±0.4	50.9±1.0
	D^2BN	65.8±1.6	**81.1±1.2**	**71.3±0.7**	**72.3±0.3**	62.5±2.0	**79.8±1.3**	**64.7±1.1**	**51.8±0.2**

Hyperparameters are set based on validation set (see below classification on how we split validation set). For ProdLDA, DVAE, and our models, we set 1 as concentration parameter for Dirichlet prior. For models with word embeddings, including ours, we use 300D GloVe. Structure embeddings $\mathbf{x}_i$ are pre-trained by DeepWalk [69]. We set $L = 3$ convolutional layers. Dropout rate is 0.6. $\lambda_{\text{OT}} = 2$ and $\lambda_s = 1$ after searching in $[0.5, 1, 2, 4, 10]$. $\lambda_{\text{KL}} = 0.001$ for KL divergence. $\gamma = 20$ for OT. Each result is obtained by 5 independent runs. We report both mean and std.dev.

7.4.1 Quantitative Evaluation

Document Classification. Documents within the same category share similar topics. As in LDA [6], we do document classification. We randomly split 80% documents for training (among which 10% are for validation), 20% for testing. During training, we observe training documents and links within them. Labels are never involved when training. After convergence, we infer topics of test documents and classify them with κNN [4]. We input topic distributions of training documents to κNN and predict the labels of test documents.

Table 7.3: Document clustering NMI (left) and Link prediction AUC (right) at $K = 64$. Results are in percentage. LANTM cannot run on Aminer and Web even on a machine with 256GB. Web dataset does not have ground-truth labels, thus can not evaluate document clustering.

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

We first vary the number of topics K and report micro f1 score with 5NN at Fig. 7.2(a-d). LANTM cannot run on large dataset Aminer even on a machine with 256GB, thus is excluded. Overall, document network models generally perform better than models with content only, since network structure indicates topic similarity among documents, and modeling it can bring similar documents closer, thus achieving better results. The improvement of our models over network baselines, especially on Aminer with short texts, verifies the advantage of pre-trained word embeddings to alleviate word sparsity. Both our models perform stably across different number of topics. D^{2}BN generally achieves better results than DBN, since double barycenter regularizes topic distributions by explicitly modeling network. Since most models begin to plateau at 64 topics, we keep $K = 64$ for subsequent experiments.

We then vary the number of nearest neighbors κ for κNN classifier. Fig. 7.2(e-h) shows the results at $K = 64$ topics. Overall, both our models outperform baselines across different number of nearest neighbors. Most models present a stable performance. GNCTM jointly models text and network structure, and is the best-performing baseline. It is competitive with DBN on ML dataset, but is still worse than D^{2}BN. LANTM extends VGAE as a topic model with Gaussian prior. The comparison to LANTM verifies the advantage of Dirichlet prior. As shown by previous

Table 7.4: Topic coherence NPMI (left, in percentage) and perplexity (right, lower is better) at $K = 64$. LANTM cannot run on Aminer and Web even on a machine with 256GB. VGAE is not a topic model and cannot evaluate topic coherence and perplexity.

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

works [8], Dirichlet forces topics to be sparser than Gaussian and achieves lower reconstruction error, thus improving classification. Table 7.2 summarizes both micro and macro f1 scores at $K = 64$.

Document Clustering. As in [107], we evaluate topic quality by document clustering. After training, we use K-Means [4] to cluster test documents. Labels are involved only for evaluating the clustering quality by Normalized Mutual Information (NMI) [85]. Table 7.3(left) shows the results. Benefiting from both text and network structure, Adjacent-Encoder presents the best results among baselines. Though DBN is competitive with Adjacent-Encoder, the advanced D^2BN still significantly outperforms the latter, since double barycenter enhances topic quality by both semantic and structure generation. The comparison to NRTM, LANTM, and VGAE verifies the advantage of Dirichlet prior, which improves topic quality by forcing topic distributions to be sparser than Gaussian.

Link Prediction. A good model should derive similar topics for potentially linked documents. As in RTM [10], given two documents, we use their topic distributions to predict if there is a link between them. We split the dataset the same as classification. We observe only training documents and links within them for training. After convergence, we infer topics of test documents, and predict links that connect two test documents. As in [107], the probability of

Table 7.5: Topic diversity TD (in percentage) at $K = 64$. LANTM cannot run on Aminer and Web even on a machine with 256GB.

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

a link is $p(e_{ij} = 1|i, j) \propto \exp(-||\mathbf{z}_i - \mathbf{z}_j||^2)$. As in [90], we use AUC as evaluation metric. Table 7.3(right) presents the results at 64 topics.LANTM cannot run on large datasets Aminer and Web, thus is excluded. Overall, our models predict links more accurately than baselines. Compared to document network models, we highlight that modeling barycenter with auxiliary knowledge can encode similar documents closely. Compared to models without networks, we verify that network indeed brings useful information.

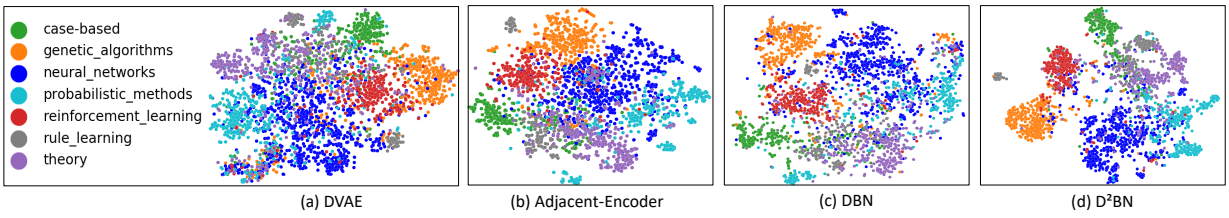


Figure 7.3: T-SNE topic visualization on ML dataset.

Table 7.6: Top-10 key words of 5 randomly selected topics on ML dataset.

Topic	Top-10 key words of DBN
1	optimization, algorithm, trade-off, constraint, scalability, optimal, tradeoff, optimisation, iterative, minimization
2	methodology, theoretical, mathematical, empirical, theory, computational, analysis, modeling, analytical, conceptual
3	multivariate, stochastic, regression, bayesian, nonlinear, parameter, dynamic, gaussian, generalization, inverse
4	generalization, inference, causality, empirical, probabilistic, completeness, causal, causation, first-order, predicate
5	visual, color, image, pattern, subtle, object, eye, contrast, characteristic, optical

Topic	Top-10 key words of D^{2}BN
1	finite, topological, symmetric, algebraic, orthogonal, invariant, generalization, topology, subset, discrete
2	feed-forward, multi-layer, feedforward, connectionism, connectionist, two-layer, self-organizing, kohonen, self-organization, network-based
3	processor, microprocessor, parallelism, simd, interface, functionality, hardware, instruction, server, workstation
4	document, copy, text, detailed, instance, specific, read, publication, book, publish
5	genetic, mutation, organism, phenotype, evolution, gene, molecular, evolutionary, trait, protein

7.4.2 Topic Analysis

Topic Coherence. An important property of topic model is semantic interpretability, i.e., each document is represented by a topic distribution, and each topic is interpreted by a group of key words. To evaluate semantic interpretability, we use topic coherence to test if the key words of a topic coherently reflect the same semantic meaning. Our decoding parameter $2 - \mathbf{C} \in \mathbb{R}^{K \times |\mathcal{V}|}$ is topic-word distribution. Each row is the distribution of a topic over the words, the key words of that topic are the highest values on that row. As in ProdLDA [76], we use NPMI to evaluate the coherence of top-10 key words of each topic. We use *Google Web 1T 5-gram Version 1* [22] for evaluation. VGAE is not a topic model and cannot evaluate coherence.

Table 7.4(left) shows that benefiting from OT, NSTM produces the most coherent topics among baselines. Compared to it, we design OT barycenters to model network structure, thus significantly improve NPMI by 2.1 on average. This is because network helps to capture the situation where two linked documents present different content but consistent semantics. DBN generally performs better than D^{2}BN, since D^{2}BN pays some optimization effort to the reconstruction of adjacency vector, which reduces the precision of content generation. But D^{2}BN is still better than most baselines.

Perplexity. A model should generalize well to unseen documents. After training, as in LDA [6], we evaluate perplexity, $\exp\{-\frac{\log \Pr(D_{test})}{\sum_{d' \in D_{test}} N_{d'}}\}$, of 20% test documents. Since perplexity is exponential and varies a lot w.r.t. its power, we instead report its power, $-\frac{\log \Pr(D_{test})}{\sum_{d' \in D_{test}} N_{d'}}$. Lower

is better. Table 7.4(right) shows that our models consistently provide higher likelihood to test documents than baselines, since network helps aggregate text of different documents to alleviate sparsity. Compared to LANTM, we attribute the improvement to Dirichlet prior, which achieves lower reconstruction error than Gaussian, thus providing a better generation to test documents.

Topic Diversity. It is important to evaluate if the discovered K different topics are diverse and not repetitive. Another commonly used metric is topic diversity [1, 63, 118], i.e., TD $= \frac{N_{\text{unique}}}{N_{\text{total}}}$, the percentage of unique key words in the top-10 key words of K topics. TD close to 1 reveals semantically diverse topics. TD close to 0 indicates a large proportion of redundant topics, and other semantically meaningful topics are unexplored, which results in inferior topic discovery.

Table 7.5 summarizes the results. RTM and Adjacent-Encoder discover the most diverse topics among baselines. Network structure connects similar documents and separates disconnected ones. Modeling network forces topic distributions to focus on each subgroup of connected documents. Different local topology of the network helps improve topic diversity. Compared to RTM and Adjacent-Encoder, we derive more diverse topics, since we use pre-trained embeddings as external knowledge, which capture more diverse and robust semantic meaning. The outperformance over LANTM verifies the advantage of Dirichlet prior.

Topic Interpretability. To intuitively understand what topics our models capture, we randomly select 5 topics and show top-10 key words on ML dataset at Table 7.6. Most key words coherently reflect the consistent topic. For example, DBN's topic 1 reveals *Constrained Optimization*, and its topic 3 discusses *Multivariate Regression*. For D^2BN, topic 2 reflects *Feed-Forward Neural Network*, topic 3 seems *Parallel Processing*, and topic 5 shows *Computational Biology*.

Topic Visualization. To visually understand how topic distributions are learned, we apply t-SNE [82] to project 64-dimensional topic distributions to 2-dimensional visual space, and color documents with their labels. Fig. 7.3 shows the results on ML. Compared to other three plots,

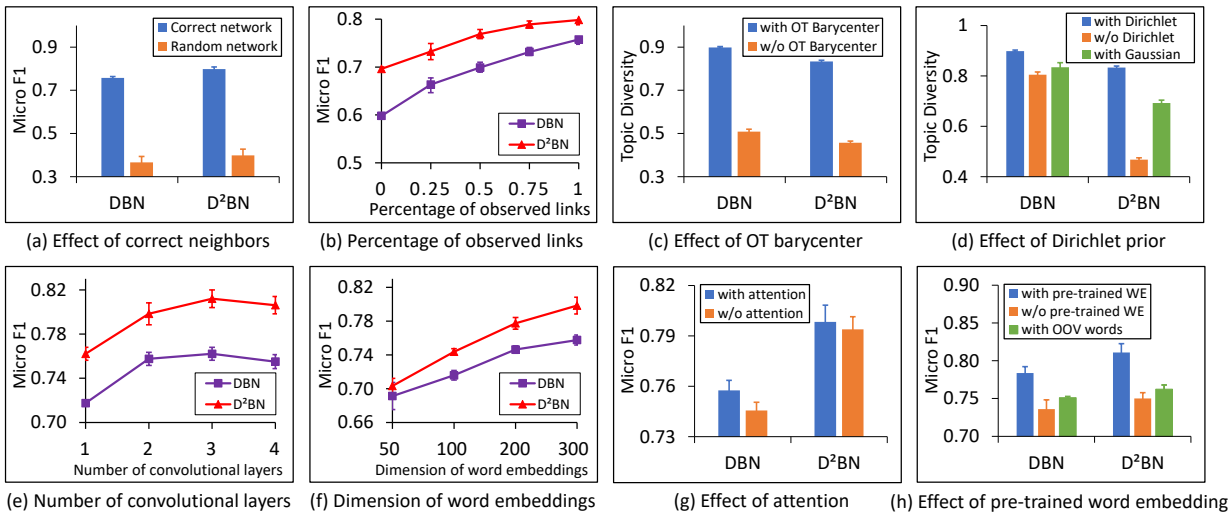


Figure 7.4: Model analysis on ML dataset.

DVAE does not model network structure, thus mixes more documents from different categories. Benefiting from document adjacency, Adjacent-Encoder and DBN present almost similar separation among categories based on visual observation. D^{2}BN produces clearer boundaries, due to double barycentric modeling.

7.4.3 Model Analysis

Effect of Network Structure. To further verify network structure indeed helps topic modeling, we conduct two experiments. ***i***) We randomly connect documents but keep the number of neighbors each document has. The new network contains wrongly connected neighbors. Fig. 7.4(a) shows that a random network brings noisy information on neighborhood and deteriorates the results. This observation enhances the advantage of a correctly connected network. ***ii***) We randomly remove a proportion of observed links and conduct document classification using the remaining links. Fig. 7.4(b) shows classification results with different percentage of observed links. Compared to the case with no links, more links are observed, better the performance. Links indeed bring useful information on document similarity, and modeling them improves the quality of learned topic distributions.

Effect of OT Barycenter. An important component is optimal transport barycenter, which captures document network and incorporates pre-trained embeddings. To test its effect, we remove it from our models and report topic diversity at Fig. 7.4(c). Both models significantly drop topic diversity without OT barycenter. One potential reason is that OT is good at measuring semantic distance. Given a topic distribution $\mathbf{z}_i$ and word distribution $\mathbf{d}_i$, OT aims to find the optimal plan between them with less transport cost. Since we define cost matrix using topic and word embeddings, and every topic involves a certain amount of cost, the minimization of OT barycenter uses as few topics as possible to achieve the transportation between topic $\mathbf{z}_i$ and word $\mathbf{d}_i$. As a result, OT barycenter pushes each topic to focus on its distinct semantics, so that a few diverse topics can still achieve a low-cost transportation. Different topics discuss distinct semantics. The minimization of OT thus produces diverse topics.

Effect of Dirichlet Prior. Previous models with optimal transport, e.g., NSTM [118], do not impose any prior on topic distribution. Motivated by LDA [6], we use rejection sampling to introduce Dirichlet as an OT prior. We conduct two experiments to evaluate its usefulness. ***i***) We first remove Dirichlet prior and do not impose any prior for topics. ***ii***) Since another widely used prior is Gaussian, adopted by our baselines ETM [19], NRTM [2], and LANTM [90], we replace Dirichlet with Gaussian. Fig. 7.4(d) illustrates that Dirichlet prior is indeed useful to discover more diverse topics than the models without prior or with Gaussian. This is because Dirichlet is able to derive sparse topic distributions, as verified by previous works [8]. The sparsity of topic distributions allows each document to focus on only a subset of distinct topics for content generation. These subset of topics gradually become specialized in certain semantics, and different subsets of topics preserve unique features. Such process finally produces diverse topics. From Fig. 7.4(d), we also conclude that having a prior is better than not having it, since the models with Gaussian prior still improve topic diversity compared to the case with no prior. Gaussian prior has zero mean, which forces topics to zero vectors. In contrast, models with no prior do not have this regularization, leading to overlapping topics.

Different Number of Convolutional Layers. We vary the number of convolutional layers L for our encoder, and summarize classification result at Fig. 7.4(e). When there is only one layer $L = 1$, our model cannot capture higher-order network connectivity, resulting in a low accuracy. After we gradually increase L to 2 and 3, we observe a significant improvement, since structural information is well encoded into document topic distributions. However, an overly high number of convlutional layer influences the result, since more noisy neighbors are encoded.

Dimension of Word Embeddings. We used 300D word embeddings for all above experiments. To investigate the effect of different dimensions, we conduct document classification and show the results at Fig. 7.4(f). Overall, when we increase the dimension of word embeddings, both models present an improving trend, since a high dimension of word embeddings can capture more semantic information, thereby boosting the results. D^2BN performs more stably than DBN, because the generation of adjacency vector constitutes another information and helps dilute the effect of word embedding dimensions.

Effect of Attention. Neighboring weights help to differentiate neighbors for barycentric modeling and decoding. To test its effect, we replace the weights with uniform values, and neighbors are equally important. We show the comparison at Fig. 7.4(g). We discover that removing weights leads to worse results, which verifies the importance of its design. DBN drops more than D^2BN without attention, since adjacency vector in D^2BN represents additional useful information to offset the influence of uniform attention.

Effect of Pre-trained Word Embeddings. To test the effect of pre-trained word embeddings, we conduct two experiments. ***i***) We replace pre-trained word embeddings with randomly initialized ones, and train word embeddings together with other parameters. ***ii***) To test if our models can handle our-of-vocabulary (OOV) words, we randomly remove 20% words. The new vocabulary contains 80% words only. Thus, training documents may contain removed words (OOV words). For OOV words, we check training documents and obtain their contextual words, and take the average of contextual word embeddings as the embedding of OOV words. Fig.

7.4(h) shows that the models with randomly initialized word embeddings drop the performance, since models do not capture auxiliary knowledge. If the corpus contains OOV words, the performance improves over the models with randomly initialized embeddings, since models still leverage partial auxiliary knowledge. The models with OOV words drop the result compared to the models with non-OOV words, since the inferred embeddings of OOV words may not be optimal. But overall, the performance of models with OOV words is decent, verifying that we can infer OOV word embeddings for training.

7.5 Discussion

We propose GNN topic models for networked documents based on Optimal Transport Barycenter. For DBN, we incorporate network structure implicitly by designing a topic barycenter. D^{2}BN enhances DBN by explicitly pushing topics to the double barycenters of both semantic and structure spaces. We inject pre-trained word embeddings into the cost matrix of optimal transport to alleviate word sparsity problem of short documents. To impose Dirichlet as an optimal transport prior, we use rejection sampling. Extensive experiments verify the effectiveness of our models.

Appendix

Proof. (***i***) We here prove the first inequality, $-\sum_{j\in\mathcal{N}(i)} a_{ij} l(\mathbf{z}_i, \mathbf{d}_j) \geq \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j)$. Its proof is similar to [118]. As a self-contained paper, we provide the details.

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

$$\hat{\phi} = \sum_{w\in\mathcal{V}} \exp(\sum_{k=1}^{K} z_{ik}(2 - c_{kw})) = e^2 \sum_{w\in\mathcal{V}} \exp(-\sum_{k=1}^{k} z_{ik} c_{kw}). \tag{7.16}$$

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

$$\begin{aligned}
&\sum_{j\in\mathcal{N}(i)} a_{ij} l(\mathbf{z}_i, \mathbf{d}_j) = \sum_{j\in\mathcal{N}(i)} \frac{a_{ij}}{\sum_{w'\in\mathcal{V}} n_{j,w'}} \sum_{w\in\mathcal{V}} n_{j,w} \log \phi(\mathbf{z}_i)_w \\
&= \sum_{j\in\mathcal{N}(i)} \frac{a_{ij}}{\sum_{w'\in\mathcal{V}} n_{j,w'}} \sum_{w\in\mathcal{V}} n_{j,w} (\sum_{k=1}^{K} z_{ik}(2 - c_{kw}) - \log \hat{\phi}) \\
&= \sum_{j\in\mathcal{N}(i)} a_{ij} \left(2 - \log \hat{\phi} - \frac{1}{\sum_{w'\in\mathcal{V}} n_{j,w'}} \sum_{w\in\mathcal{V}} n_{j,w} \sum_{k=1}^{K} z_{ik} c_{kw} \right)
\end{aligned} \tag{7.17}$$

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

$$\begin{aligned}
&\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) = \sum_{j\in\mathcal{N}(i)} a_{ij} \min_{\mathbf{P}\in U'(\mathbf{z}_i,\mathbf{d}_j)} \sum_{w\in\mathcal{V}} \sum_{k=1}^{K} p_{kw} d_{jw} c_{kw} \\
&= \sum_{j\in\mathcal{N}(i)} \frac{a_{ij}}{\sum_{w'\in\mathcal{V}} n_{jw'}} \min_{\mathbf{P}\in U'(\mathbf{z}_i,\mathbf{d}_j)} \sum_{w\in\mathcal{V}} n_{jw} \sum_{k=1}^{K} p_{kw} c_{kw}.
\end{aligned} \tag{7.18}$$

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

$$\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \leq \sum_{j\in\mathcal{N}(i)} \frac{a_{ij}}{\sum_{w'\in\mathcal{V}} n_{jw'}} \sum_{w\in\mathcal{V}} n_{jw} \sum_{k=1}^{K} z_{ik} c_{kw}. \tag{7.19}$$

Taking Eq. 7.17 and 7.19 together, given $|\mathcal{V}| \geq 8$, we have

$$\begin{aligned}
-\sum_{j\in\mathcal{N}(i)} a_{ij} l(\mathbf{z}_i, \mathbf{d}_j) &= -\sum_{j\in\mathcal{N}(i)} a_{ij}\Big(2 - \log\hat{\phi} \\
&\qquad - \frac{1}{\sum_{w'\in\mathcal{V}} n_{j,w'}} \sum_{w\in\mathcal{V}} n_{j,w} \sum_{k=1}^{K} z_{ik} c_{kw}\Big) \\
&\geq -2 + \log\hat{\phi} + \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \\
&= \log \sum_{w\in\mathcal{V}} \exp(-\sum_{k=1}^{k} z_{ik} c_{kw}) + \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \\
&\geq \log(|\mathcal{V}|) - 2 + \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \\
&\geq \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j).
\end{aligned} \tag{7.20}$$

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

We apply the proof by contradiction. Given $\mathbf{C}$ and a_{ij}, we assume $\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) < \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_j, \mathbf{d}_j)$ is correct, which tells us that the solution Eq. 7.21 is better than Eq. 7.22. Here we use $\mathbf{z}_j^*$ to represent the best solution of OT $d_{\mathbf{C}}(\cdot, \mathbf{d}_j)$.

$$\mathbf{z}_1^* = \mathbf{z}_2^* = ... = \mathbf{z}_{|\mathcal{N}(i)|}^* = \mathbf{z}_i \tag{7.21}$$

$$\mathbf{z}_1^* = \mathbf{z}_1, \quad \mathbf{z}_2^* = \mathbf{z}_2, \quad ..., \quad \mathbf{z}_{|\mathcal{N}(i)|}^* = \mathbf{z}_{|\mathcal{N}(i)|} \tag{7.22}$$

Eq. 7.21 shows that all the best solutions are the same, while Eq. 7.22 shows that each individual OT has its own solution and different OTs may produce different solutions. However, given that $\mathbf{z}_j = \arg\min_{\mathbf{z}_j} d_{\mathbf{C}}(\mathbf{z}_j, \mathbf{d}_j)$ is already the best solution of its individual OT, other solutions, such as $\mathbf{z}_j^* = \mathbf{z}_i$, may not produce the minimum OT distance. Equivalently, the assumption of $\sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) < \sum_{j\in\mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_j, \mathbf{d}_j)$ contradicts the given information.

Based on the analysis above, we have

$$d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \geq d_{\mathbf{C}}(\mathbf{z}_j, \mathbf{d}_j). \tag{7.23}$$

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

$$\sum_{j \in \mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_i, \mathbf{d}_j) \geq \sum_{j \in \mathcal{N}(i)} a_{ij} d_{\mathbf{C}}(\mathbf{z}_j, \mathbf{d}_j). \tag{7.24}$$

□

Chapter 8

Conclusion and Future Work

In this dissertation, we explore the document graph representation learning problem. We seek to develop topic models that can incorporate both textual content and network connectivity. Based on the observation of the temporally growing document network, we model topic evolution to capture dynamic process of the corpus. In order to model associated authors and venues of a document, we extend Variational Graph Auto-Encoder and design a hierarchical multi-layered document graph to incorporate auxiliary authors and venues. We also alleviate the problem of short documents by designing a meta-learning method and optimal transport barycenter to incorporate pre-trained word embeddings, respectively.

In Chapter 3, we propose *Adjacent-Encoder* and *Adjacent-Encoder-X*, neural topic models that learn unified representations for networked documents. *Adjacent-Encoder* incorporates the network structure implicitly, with similar number of parameters as Auto-Encoder family, yet outperforms the latter. *Adjacent-Encoder-X* that models the network structure explicitly performs even better. Empirical analysis on public datasets support these findings, showcasing the effectiveness of factoring network structure for neural topic modeling. The model extensions, such as denoising, contractive, and sparsity, further improve the performance.

To model dynamic process of a growing document network, in Chapter 4, we propose two neural topic models for dynamic document networks, which are notable in jointly preserving dy-

namicity and network adjacency. By designing a time-aware optimal transport, NetDTM models each link by semantically generating content of neighbors. NetDTM++ further extends NetDTM to incorporate the effect of historical links by a Hawkes process. Experiments on several dynamic document networks covering academic literature and Web documents show the effectiveness of our models against baselines on various aspects, including deriving latent document representations more amenable to classification and link prediction metrics as well as modeling the evolution of topics.

Authors and venues represent auxiliary information associated with documents, which reveal the similarity of documents through author and venue connectivity. We extend Variational Graph Auto-Encoder and design a hierarchical multi-layered document graph at Chapter 5 for authorship and venue modeling. In order to achieve a promising topic modeling quality, we design three word relations (contextual, syntactic, and semantic) for word layers. We also investigate three alternatives of variational divergence term.

Short text topic modeling is an important research direction in topic modeling, since scarce word co-occurrences influence the accurate topic discovery process. Chapter 6 solves this problem by introducing meta-learning into topic modeling to transfer semantic knowledge learned on long documents to complement the word scarcity of short texts in a self-contained manner, so that no additional auxiliary information is needed.

Chapter 7 approaches short text topic modeling from another perspective. We propose two neural topic models for networked documents based on the concept of Optimal Transport Barycenter, which benefits our models by naturally incorporating pre-trained word embeddings to alleviate short text problem. We incorporate network structure implicitly by designing a topic barycenter for DBN. D2BN enhances DBN by explicitly pushing topics to the dual barycenter of both semantic and structure spaces.

Future Work. My research focuses on text mining and graph representation learning. For my future research directions, I will continue focusing on these related areas and involve multiple

different real-world scenarios.

One potential research direction is to use pre-trained large language models for text-based recommender systems. On the one hand, pre-trained large language models have achieved promising performance for text modeling, but its design in recommender systems is still unexplored. Recommender systems, on the other hand, usually involve a user-item interaction graph structure, where the links between users and items sometimes couple with textual reviews, which are written by the user for the corresponding item. We can convert such an e-commerce scenario to a type of document graph. It is possible to integrate pre-trained large language model and graph representation learning techniques into a unified model to build recommender systems.

Another research direction is to explore the scenario of chemical topic modeling area. In chemical research, learning molecule embedding is always an important research direction, since the learned embeddings may produce undiscovered chemical reactions based on the similarity between relevant molecules. Furthermore, each molecule is a graph structure where atoms, representing nodes on the graph, are connected through chemical keys, corresponding to graph links. Existing research mainly focuses on how to leverage molecule graph structure and chemical reactions to improve molecule representations. However, if a molecule does not have a strongly connected graph structure or if it lacks sufficient observed chemical reactions, the embedding quality may be influenced due to scarce observed information. We discover that textual description of molecules can be used as additional information to improve molecule representations, since molecules with similar chemical properties should have overlapping textual content. Therefore, a promising future research direction is to unify text with molecule to improve representation quality, meanwhile predicting chemical property of a new molecule using the power of generative topic modeling.

Bibliography

[1] AZARBONYAD, H., DEHGHANI, M., KENTER, T., MARX, M., KAMPS, J., AND DE RIJKE, M. Hitr: Hierarchical topic model re-estimation for measuring topical diversity of documents. *IEEE Transactions on Knowledge and Data Engineering 31*, 11 (2018), 2124–2137. 7.4.2

[2] BAI, H., CHEN, Z., LYU, M. R., KING, I., AND XU, Z. Neural relational topic models for scientific article analysis. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (2018), pp. 27–36. 1.2, 1.2, 2.2, 3.4.1, 4.1, 4.4, 2, 7.4.3

[3] BHADURY, A., CHEN, J., ZHU, J., AND LIU, S. Scaling up dynamic topic models. In *Proceedings of the 25th International Conference on World Wide Web* (2016), pp. 381–390. 2.2

[4] BISHOP, C. M. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995. 1.2, 2.2, 3.5, 4.4.1, 6.4.1, 7.3.1, 7.4.1

[5] BLEI, D. M., AND LAFFERTY, J. D. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning* (2006), pp. 113–120. 1.2, 2.2, 4.1, 4.1, 4.3.1, 4.4, 4.4.2

[6] BLEI, D. M., NG, A. Y., AND JORDAN, M. I. Latent dirichlet allocation. *the Journal of machine Learning research 3* (2003), 993–1022. 1.2, 1.2, 1.2, 2.2, 4.4, 4.4.1, 4.4.2, 5.1, 5.4.3, 5.5.1, 5.5.2, 6.3.3, 6.4.1, 6.4.1, 7.1, 7.3.1, 7.3.2, 7.3.2, 7.4.1, 7.4.2, 7.4.3

[7] BOUMA, G. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL 30* (2009), 31–40. 3.4.4

[8] BURKHARDT, S., AND KRAMER, S. Decoupling sparsity and smoothness in the dirichlet variational autoencoder topic model. *Journal of Machine Learning Research 20*, 131 (2019), 1–27. 2.2, 4.3.1, 4.3.1, 5.5, 5.5.1, 7.3.1, 7.3.1, 7.3.1, 1, 7.4.1, 7.4.3

[9] CAO, S., LU, W., AND XU, Q. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management* (2015), pp. 891–900. 2.1

[10] CHANG, J., AND BLEI, D. Relational topic models for document networks. In *Artificial intelligence and statistics* (2009), PMLR, pp. 81–88. 1.2, 1.2, 1.2, 2.2, 3.1, 3.4.1, 4.1, 4.1, 4.4, 4.4.1, 5.5, 5.5.1, 6.4, 6.4.1, 2, 7.4.1

[11] CHEN, C., TONG, H., XIE, L., YING, L., AND HE, Q. Fascinate: fast cross-layer depen-

dency inference on multi-layered networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), pp. 765–774. 2.1

[12] CHEN, C., TONG, H., XIE, L., YING, L., AND HE, Q. Cross-dependency inference in multi-layered networks: A collaborative filtering perspective. *ACM Transactions on Knowledge Discovery from Data (TKDD) 11*, 4 (2017), 1–26. 2.1

[13] CHEN, Y., AND ZAKI, M. J. Kate: K-competitive autoencoder for text. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2017), pp. 85–94. 2.2, 3.3.1, 3.4.1

[14] CUTURI, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems 26* (2013), 2292–2300. 1.2, 1.2, 2.2, 4.3.1, 7.3.4

[15] CUTURI, M., AND DOUCET, A. Fast computation of wasserstein barycenters. In *International conference on machine learning* (2014), PMLR, pp. 685–693. 1.2

[16] DA XU, CHUANWEI RUAN, EVREN KORPEOGLU, SUSHANT KUMAR, AND KANNAN ACHAN. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations* (2020). 2.1, 4.3.1

[17] DAS, R., ZAHEER, M., AND DYER, C. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (2015), pp. 795–804. 2.2, 5.3.2, 6.3.4

[18] DIENG, A. B., RUIZ, F. J., AND BLEI, D. M. The dynamic embedded topic model. *arXiv preprint arXiv:1907.05545* (2019). 2.2, 4.3.1, 4.4

[19] DIENG, A. B., RUIZ, F. J., AND BLEI, D. M. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics 8* (2020), 439–453. 1.2, 2.2, 6.4, 1, 7.4.3

[20] DING, K., WANG, J., LI, J., LI, D., AND LIU, H. 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)* (2020), pp. 4927–4936. 1.2, 2.1, 5.5

[21] DONG, Y., CHAWLA, N. V., AND SWAMI, A. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (2017), pp. 135–144. 2.1

[22] EVERT, S. Google web 1t 5-grams made easy (but not for the computer). In *Proceedings of the NAACL HLT 2010 Sixth Web as Corpus Workshop* (2010), pp. 32–40. 3.4.4, 4.4.2, 5.5.2, 6.4.1, 7.4.2

[23] FINN, C., ABBEEL, P., AND LEVINE, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning* (2017), PMLR, pp. 1126–1135. 1.2, 2.2, 6.2

[24] FINN, C., XU, K., AND LEVINE, S. Probabilistic model-agnostic meta-learning. *Advances in neural information processing systems 31* (2018). 2.2

[25] FU, T.-Y., LEE, W.-C., AND LEI, Z. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (2017), pp. 1797–1806. 2.1

[26] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A., AND BENGIO, Y. Generative adversarial nets. *Advances in neural information processing systems 27* (2014). 2.1

[27] GROVER, A., AND LESKOVEC, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (2016), pp. 855–864. 2.1

[28] GROVER, A., ZWEIG, A., AND ERMON, S. Graphite: Iterative generative modeling of graphs. In *International conference on machine learning* (2019), PMLR, pp. 2434–2444. 2.1

[29] HAMILTON, W. L., YING, R., AND LESKOVEC, J. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (2017), pp. 1025–1035. 2.1

[30] HOFMANN, T. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (1999), pp. 50–57. 1.2, 4.4.2

[31] HU, B., FANG, Y., AND SHI, C. Adversarial learning on heterogeneous information networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019), pp. 120–129. 2.1

[32] HU, W., AND TSUJII, J. A latent concept topic model for robust topic inference using word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (2016), pp. 380–386. 2.2

[33] HUYNH, V., ZHAO, H., AND PHUNG, D. Otlda: A geometry-aware optimal transport approach for topic modeling. *Advances in Neural Information Processing Systems 33* (2020). 1.2, 1.2, 1.2, 2.2

[34] IWATA, T., YAMADA, T., SAKURAI, Y., AND UEDA, N. Online multiscale dynamic topic models. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (2010), pp. 663–672. 2.2

[35] JÄHNICHEN, P., WENZEL, F., KLOFT, M., AND MANDT, S. Scalable generalized dynamic topic models. In *International Conference on Artificial Intelligence and Statistics* (2018), PMLR, pp. 1427–1435. 2.2

[36] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015). 4.3.2

[37] KINGMA, D. P., MOHAMED, S., REZENDE, D. J., AND WELLING, M. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems* (2014), pp. 3581–3589. 2.2

[38] KINGMA, D. P., AND WELLING, M. Auto-encoding variational bayes. *arXiv preprint*

arXiv:1312.6114 (2013). 2.1, 2.2, 3.4.1, 4.4, 5.4.2

[39] KIPF, T. N., AND WELLING, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*. 2.1, 5.2.1, 5.4.2

[40] KIPF, T. N., AND WELLING, M. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning* (2016). 1.2, 2.1, 3.4.1, 5.1, 5.4.1, 5.4.3, 5.5, 7.1, 7.3, 3

[41] LACOSTE-JULIEN, S., SHA, F., AND JORDAN, M. I. Disclda: Discriminative learning for dimensionality reduction and classification. In *Advances in neural information processing systems* (2009), pp. 897–904. 2.2

[42] LE, T. M., AND LAUW, H. W. Probabilistic latent document network embedding. In *2014 IEEE International Conference on Data Mining* (2014), IEEE, pp. 270–279. 2.2, 3.4.1, 3.4.2, 4.4.1

[43] LECUN, Y., BENGIO, Y., ET AL. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks 3361*, 10 (1995), 1995. 2.1

[44] LEE, Y., AND CHOI, S. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning* (2018), PMLR, pp. 2927–2936. 2.2

[45] LESKOVEC, J., BACKSTROM, L., AND KLEINBERG, J. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (2009), pp. 497–506. 4.4, 5.5, 6.4, 7.4

[46] LESKOVEC, J., KLEINBERG, J., AND FALOUTSOS, C. 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* (2005), pp. 177–187. 4.4, 5.5, 6.4

[47] LI, C., WANG, H., ZHANG, Z., SUN, A., AND MA, Z. Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (2016), pp. 165–174. 2.2

[48] LI, J., CHEN, C., TONG, H., AND LIU, H. Multi-layered network embedding. In *Proceedings of the 2018 SIAM International Conference on Data Mining* (2018), SIAM, pp. 684–692. 2.1

[49] LI, J., YU, J., LI, J., ZHANG, H., ZHAO, K., RONG, Y., CHENG, H., AND HUANG, J. Dirichlet graph variational autoencoder. *Advances in Neural Information Processing Systems 33* (2020). 2.1

[50] LIM, K. W., AND BUNTINE, W. Bibliographic analysis with the citation network topic model. In *Asian conference on machine learning* (2015), PMLR, pp. 142–158. 2.2

[51] LIU, Q., ALLAMANIS, M., BROCKSCHMIDT, M., AND GAUNT, A. L. Constrained graph variational autoencoders for molecule design. In *Advances in Neural Information Processing Systems* (2018), pp. 7806–7815. 2.1

[52] LIU, X., YOU, X., ZHANG, X., WU, J., AND LV, P. Tensor graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*

(2020), vol. 34, pp. 8409–8416. 5.3.2

[53] LIU, Z., ZHANG, W., FANG, Y., ZHANG, X., AND HOI, S. C. Towards locality-aware meta-learning of tail node embeddings on networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (2020), pp. 975–984. 6.4

[54] LU, Y., WANG, X., SHI, C., YU, P. S., AND YE, Y. Temporal network embedding with micro-and macro-dynamics. In *Proceedings of the 28th ACM international conference on information and knowledge management* (2019), pp. 469–478. 2.1, 4.4

[55] MAKHZANI, A., AND FREY, B. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663* (2013). 2.2, 3.4.1

[56] MANNING, C. D., SURDEANU, M., BAUER, J., FINKEL, J. R., BETHARD, S., AND MCCLOSKY, D. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations* (2014), pp. 55–60. 5.3.2

[57] MCAULIFFE, J. D., AND BLEI, D. M. Supervised topic models. In *Advances in neural information processing systems* (2008), pp. 121–128. 2.2

[58] MCCALLUM, A. K., NIGAM, K., RENNIE, J., AND SEYMORE, K. Automating the construction of internet portals with machine learning. *Information Retrieval 3*, 2 (2000), 127–163. 3.4.1, 4.4, 5.5, 6.4, 7.4

[59] MEI, Q., CAI, D., ZHANG, D., AND ZHAI, C. Topic modeling with network regularization. In *Proceedings of the 17th international conference on World Wide Web* (2008), pp. 101–110. 1.2, 2.2

[60] MIAO, Y., YU, L., AND BLUNSOM, P. Neural variational inference for text processing. In *International conference on machine learning* (2016), PMLR, pp. 1727–1736. 2.2, 7.3.2

[61] MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G., AND DEAN, J. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems* (2013). 1.2, 1.2, 2.2, 4.3.1, 5.4.4, 6.3.3, 6.3.4, 7.3.2

[62] NAESSETH, C., RUIZ, F., LINDERMAN, S., AND BLEI, D. Reparameterization gradients through acceptance-rejection sampling algorithms. In *Artificial Intelligence and Statistics* (2017), PMLR, pp. 489–498. 7.3.1, 7.3.1

[63] NAN, F., DING, R., NALLAPATI, R., AND XIANG, B. Topic modeling with wasserstein autoencoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019), pp. 6345–6381. 2.2, 4.4, 5.4.3, 5.5, 6.4, 1, 7.4.2

[64] NI, J., TONG, H., FAN, W., AND ZHANG, X. Inside the atoms: ranking on a network of networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (2014), pp. 1356–1365. 2.1

[65] PAN, S., HU, R., LONG, G., JIANG, J., YAO, L., AND ZHANG, C. Adversarially regularized graph autoencoder for graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (2018), pp. 2609–2615. 2.1

[66] PATRINI, G., VAN DEN BERG, R., FORRE, P., CARIONI, M., BHARGAV, S., WELLING, M., GENEWEIN, T., AND NIELSEN, F. Sinkhorn autoencoders. In *Uncertainty in Artificial Intelligence* (2020), PMLR, pp. 733–743. 2.2

[67] PAUL, M. Cross-collection topic models: Automatically comparing and contrasting text. *Urbana 51* (2009), 61801. 2.2

[68] PENNINGTON, J., SOCHER, R., AND MANNING, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543. 1.2, 1.2, 2.2, 4.3.1, 5.3.2, 6.3.4, 7.3.2

[69] PEROZZI, B., AL-RFOU, R., AND SKIENA, S. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (2014), pp. 701–710. 2.1, 3.5, 7.3.3, 7.4

[70] RAMAGE, D., HALL, D., NALLAPATI, R., AND MANNING, C. D. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1* (2009), Association for Computational Linguistics, pp. 248–256. 2.2

[71] RAMAGE, D., MANNING, C. D., AND DUMAIS, S. Partially labeled topic models for interpretable text mining. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (2011), pp. 457–465. 2.2

[72] RIFAI, S., VINCENT, P., MULLER, X., GLOROT, X., AND BENGIO, Y. Contractive auto-encoders: Explicit invariance during feature extraction. In *ICML* (2011). 2.2, 3.4.1

[73] ROSEN-ZVI, M., GRIFFITHS, T., STEYVERS, M., AND SMYTH, P. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence* (2004), pp. 487–494. 1.2, 2.2, 5.1, 5.5

[74] SHI, C., HU, B., ZHAO, W. X., AND PHILIP, S. Y. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering 31*, 2 (2018), 357–370. 2.1

[75] SNELL, J., SWERSKY, K., AND ZEMEL, R. Prototypical networks for few-shot learning. *Advances in neural information processing systems 30* (2017). 2.2

[76] SRIVASTAVA, A., AND SUTTON, C. Autoencoding variational inference for topic models. *ICLR* (2017). 1.2, 2.2, 3.4.1, 4.4, 4.4.2, 5.4.3, 5.5, 5.5.2, 6.1, 6.3.3, 6.4, 6.4.1, 1, 7.4.2

[77] TANG, J., QU, M., WANG, M., ZHANG, M., YAN, J., AND MEI, Q. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web* (2015), pp. 1067–1077. 2.1, 4.3.2

[78] TANG, J., ZHANG, J., YAO, L., LI, J., ZHANG, L., AND SU, Z. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (2008), pp. 990–998. 1.2, 2.2, 5.5, 7.4

[79] TKACHENKO, M., AND LAUW, H. W. Comparelda: A topic model for document comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. 33,

pp. 7112–7119. 2.2

[80] TOLSTIKHIN, I., BOUSQUET, O., GELLY, S., AND SCHOELKOPF, B. Wasserstein auto-encoders. In *International Conference on Learning Representations*. 2.2

[81] TU, Y., JOHRI, N., ROTH, D., AND HOCKENMAIER, J. Citation author topic model in expert search. In *Coling 2010: Posters* (2010), pp. 1265–1273. 2.2

[82] VAN DER MAATEN, L., AND HINTON, G. Visualizing data using t-sne. *Journal of machine learning research 9*, 11 (2008). 3.4.5, 7.4.2

[83] VELIČKOVIĆ, P., CUCURULL, G., CASANOVA, A., ROMERO, A., LIÒ, P., AND BENGIO, Y. Graph attention networks. In *International Conference on Learning Representations* (2018). 2.1

[84] VINCENT, P., LAROCHELLE, H., LAJOIE, I., BENGIO, Y., MANZAGOL, P.-A., AND BOTTOU, L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research 11*, 12 (2010). 2.2, 3.4.1

[85] VINH, N. X., EPPS, J., AND BAILEY, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research 11* (2010), 2837–2854. 7.4.1

[86] VINYALS, O., BLUNDELL, C., LILLICRAP, T., WIERSTRA, D., ET AL. Matching networks for one shot learning. *Advances in neural information processing systems 29* (2016). 2.2

[87] VUORIO, R., SUN, S.-H., HU, H., AND LIM, J. J. Multimodal model-agnostic meta-learning via task-aware modulation. *Advances in Neural Information Processing Systems 32* (2019). 2.2

[88] WANG, C., BLEI, D., AND HECKERMAN, D. Continuous time dynamic topic models. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence* (2008), pp. 579–586. 2.2

[89] WANG, X., JI, H., SHI, C., WANG, B., YE, Y., CUI, P., AND YU, P. S. Heterogeneous graph attention network. In *The World Wide Web Conference* (2019), pp. 2022–2032. 2.1, 5.5

[90] WANG, Y., LI, X., AND OUYANG, J. Layer-assisted neural topic modeling over document networks. 2.1, 5.1.1, 5.5, 5.5.1, 6.4, 2, 7.4.1, 7.4.3

[91] WANG, Z., WANG, C., ZHANG, H., DUAN, Z., ZHOU, M., AND CHEN, B. Learning dynamic hierarchical topic graph with graph convolutional network for document classification. In *International Conference on Artificial Intelligence and Statistics* (2020), PMLR, pp. 3959–3969. 1.2, 2.1

[92] WU, M., AND GOODMAN, N. Multimodal generative models for scalable weakly-supervised learning. In *Advances in Neural Information Processing Systems* (2018), pp. 5575–5585. 2.2

[93] XIE, Q., HUANG, J., DU, P., AND PENG, M. Graph relational topic model with higher-order graph attention auto-encoders. In *Findings of the Association for Computational*

Linguistics: ACL-IJCNLP 2021 (2021), pp. 2604–2613. 2.1, 2

[94] XIE, Q., HUANG, J., DU, P., PENG, M., AND NIE, J.-Y. Graph topic neural network for document representation. In *Proceedings of the Web Conference 2021* (2021), pp. 3055–3065. 2.1, 2

[95] XIE, Q., HUANG, J., DU, P., PENG, M., AND NIE, J.-Y. 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* (2021), pp. 4218–4227. 2.1, 5.4.2, 5.5

[96] XIE, Q., ZHU, Y., HUANG, J., DU, P., AND NIE, J.-Y. Graph neural collaborative topic model for citation recommendation. *ACM Transactions on Information Systems (TOIS) 40*, 3 (2021), 1–30. 2.1, 2

[97] XU, H., LUO, D., HENAO, R., SHAH, S., AND CARIN, L. Learning autoencoders with relational regularization. In *International Conference on Machine Learning* (2020), PMLR, pp. 10576–10586. 1

[98] XU, H., WANG, W., LIU, W., AND CARIN, L. Distilled wasserstein learning for word embedding and topic modeling. *Advances in Neural Information Processing Systems* (2018). 1.2, 2.2

[99] XU, K., HU, W., LESKOVEC, J., AND JEGELKA, S. How powerful are graph neural networks? In *International Conference on Learning Representations* (2018). 2.1

[100] YANG, L., WU, F., GU, J., WANG, C., CAO, X., JIN, D., AND GUO, Y. Graph attention topic modeling network. In *Proceedings of The Web Conference 2020* (2020), pp. 144–154. 1.2, 1.2, 2.1, 6.1, 6.3.1, 6.3.3, 6.4

[101] YAO, H., WEI, Y., HUANG, J., AND LI, Z. Hierarchically structured meta-learning. In *International Conference on Machine Learning* (2019), PMLR, pp. 7045–7054. 2.2

[102] YAO, H., WU, X., TAO, Z., LI, Y., DING, B., LI, R., AND LI, Z. Automated relational meta-learning. In *International Conference on Learning Representations* (2019). 2.2

[103] YAO, L., MAO, C., AND LUO, Y. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence* (2019), vol. 33, pp. 7370–7377. 1.2, 2.1, 5.1, 5.3.2, 5.5

[104] YOON, J., KIM, T., DIA, O., KIM, S., BENGIO, Y., AND AHN, S. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems* (2018), vol. 31. 2.2

[105] YOU, J., YING, R., AND LESKOVEC, J. Position-aware graph neural networks. In *International Conference on Machine Learning* (2019), PMLR, pp. 7134–7143. 2.1

[106] ZHAI, C., VELIVELLI, A., AND YU, B. A cross-collection mixture model for comparative text mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (2004), pp. 743–748. 2.2

[107] ZHANG, C., AND LAUW, H. W. Topic modeling on document networks with adjacent-encoder. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020), vol. 34, pp. 6737–6745. 2.2, 5.5, 5.5.1, 5.5.1, 6.4, 6.4.1, 2, 7.4.1

[108] ZHANG, C., AND LAUW, H. W. Topic modeling on document networks with adjacent-encoder. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020), vol. 34, pp. 6737–6745. 4.4, 4.4.1

[109] ZHANG, D. C., AND LAUW, H. W. Meta-complementing the semantics of short texts in neural topic models. In *Advances in Neural Information Processing Systems*. 2.1

[110] ZHANG, D. C., AND LAUW, H. W. Representation learning on multi-layered heterogeneous network. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2021), Springer, pp. 399–416. 2.1

[111] ZHANG, D. C., AND LAUW, H. W. Semi-supervised semantic visualization for networked documents. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2021), Springer, pp. 762–778. 2.2

[112] ZHANG, D. C., AND LAUW, H. W. Topic modeling for multi-aspect listwise comparisons. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021), pp. 2507–2516. 2.2

[113] ZHANG, D. C., AND LAUW, H. W. Dynamic topic models for temporal document networks. In *Proceedings of the 39th International Conference on Machine Learning* (17–23 Jul 2022), vol. 162 of *Proceedings of Machine Learning Research*, PMLR, pp. 26281–26292. 2.2, 2

[114] ZHANG, D. C., AND LAUW, H. W. Variational graph author topic modeling. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022), pp. 2429–2438. 2.2

[115] ZHANG, H., CHEN, B., GUO, D., AND ZHOU, M. Whai: Weibull hybrid autoencoding inference for deep topic modeling. *ICLR* (2018). 2.2

[116] ZHANG, Y., YU, X., CUI, Z., WU, S., WEN, Z., AND WANG, L. Every document owns its structure: Inductive text classification via graph neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020), pp. 334–339. 2.1

[117] ZHAO, H., DU, L., BUNTINE, W., AND LIU, G. Metalda: A topic model that efficiently incorporates meta information. In *2017 IEEE International Conference on Data Mining (ICDM)* (2017), IEEE, pp. 635–644. 2.2

[118] ZHAO, H., PHUNG, D., HUYNH, V., LE, T., AND BUNTINE, W. Neural topic model via optimal transport, 2020. 1.2, 1.2, 2.2, 4.3.1, 4.3.1, 4.4, 5.5, 6.3.4, 6.4, 7.3.2, 7.3.2, 1, 7.4.2, 7.4.3, 7.5

[119] ZHOU, D., HU, X., AND WANG, R. Neural topic modeling by incorporating document relationship graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2020), pp. 3790–3796. 2.2, 1

[120] ZHU, J., AHMED, A., AND XING, E. P. Medlda: maximum margin supervised topic models. *the Journal of machine Learning research 13*, 1 (2012), 2237–2278. 2.2

[121] ZHU, Q., FENG, Z., AND LI, X. Graphbtm: Graph enhanced autoencoded variational inference for biterm topic model. In *Proceedings of the 2018 conference on empirical*

methods in natural language processing (2018), pp. 4663–4672. 2.1

[122] ZHU, S., YU, K., CHI, Y., AND GONG, Y. Combining content and link for classification using matrix factorization. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (2007), pp. 487–494. 3.4.1, 4.4

[123] ZUO, Y., LIU, G., LIN, H., GUO, J., HU, X., AND WU, J. Embedding temporal network via neighborhood formation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (2018), pp. 2857–2866. 2.1