Deep Autoencoding Topic Model with Scalable Hybrid Bayesian Inference

Hao Zhang, Bo Chen, *Senior member, IEEE,* Yulai Cong, Dandan Guo, Hongwei Liu, *Member, IEEE,* and Mingyuan Zhou

***Abstract*—To build a flexible and interpretable model for document analysis, we develop deep autoencoding topic model (DATM) that uses a hierarchy of gamma distributions to construct its multi-stochastic-layer generative network. In order to provide scalable posterior inference for the parameters of the generative network, we develop topic-layer-adaptive stochastic gradient Riemannian MCMC that jointly learns simplex-constrained global parameters across all layers and topics, with topic and layer specific learning rates. Given a posterior sample of the global parameters, in order to efficiently infer the local latent representations of a document under DATM across all stochastic layers, we propose a Weibull upward-downward variational encoder that deterministically propagates information upward via a deep neural network, followed by a Weibull distribution based stochastic downward generative model. To jointly model documents and their associated labels, we further propose supervised DATM that enhances the discriminative power of its latent representations. The efficacy and scalability of our models are demonstrated on both unsupervised and supervised learning tasks on big corpora.**

***Index Terms*—Deep topic model, Bayesian inference, SG-MCMC, document classification, feature extraction.**

I. Introduction

TO analyze a collection of documents with high-dimensional, spare, and over-dispersed bag-of-words representation, a common task is to perform topic modeling that extracts topics and topic proportions in an unsupervised manner, and another common task is to perform supervised topic modeling that jointly models the documents and their labels. While latent Dirichlet allocation (LDA) [1] and a variety of its extensions, such as these for capturing correlation structure [2], inferring the number of topics [3], [4], document categorization [5], multimodal learning [6], [7], collaborative filtering [8], and scalable inference [9], have been widely used for document analysis, the representation power of these shallow probabilistic generative models is constrained by having only a single stochastic hidden layer.

B. Chen acknowledges the support of the Program for Young Thousand Talent by Chinese Central Government, the 111 Project (No. B18039), and NSFC (61771361)

H. Liu acknowledges the support of NSFC for Distinguished Young Scholars (61525105) and Shaanxi Innovation Team Project.

M. Zhou acknowledges the support of Award IIS-1812699 from the U.S. National Science Foundation.

H. Zhang, B. Chen, Yu. Cong, D. Guo and H. Liu are with National Lab of Radar Signal Processing, Collaborative Innovation Center of Information Sensing and Understanding, Xidian University, Xi'an, Shaanxi 710071, China.

M. Zhou is with McCombs School of Business, The University of Texas at Austin, Austin, TX 78712, USA.

Corresponding author: Bo Chen, bchen@mail.xidian.edu.cn.

E-mail: zhanghao_xidian@163.com, bchen@mail.xidian.edu.cn, yulaicong@gmail.com, gdd_xidian@126.com, hwliu@xidian.edu.cn, mingyuan.zhou@mccombs.utexas.edu.

A. Related work

1) Deep generative models for documents: To address the constraint of a shallow generative model, there is a surge of research interest in multilayer representation learning for documents. To analyze the term-document count matrix of a text corpus, Srivastava et al. [10] extend a deep Boltzmann machine (DBM) with the replicated softmax topic model [11] to infer a multilayer representation with binary hidden units, but its inference network is not trained to match the true posterior [12] and the higher-layer neurons learned by DBM are difficult to visualize. Deep Poisson factor analysis (DPFA) [13] is introduced to generalize Poisson factor analysis [14], with a deep structure restricted to model binary topic usage patterns. Extending LDA, a hierarchical LDA (hLDA) is developed based on the nested Chinese restaurant process [15], in which the topics are arranged in an L-level tree and a document draws its words from a mixture of L topics within a document-specific root-to-leaf path. Deep exponential families (DEF) [16] construct more general probabilistic deep networks with non-binary hidden units, in which a count matrix can be factorized under the Poisson likelihood, with the gamma distributed hidden units of adjacent layers linked via the gamma scale parameters. The Poisson gamma belief network (PGBN) [17], [18] also factorizes a count matrix under the Poisson likelihood, but factorizes the shape parameters of the gamma distributed hidden units of each layer into the product of a connection weight matrix and the gamma hidden units of the next layer, resulting in strong nonlinearity and readily interpretable multilayer latent representations. However, the inference of PGBN in Zhou et al. [17] is based on Gibbs sampling, making it hard to be applied to big corpora, slow in out-of-sample prediction, and difficult to be jointly trained with a downstream task.

2) Scalable inference: These multilayer probabilistic models are often characterized by a top-down generative structure, with the distribution of a hidden layer typically acting as a prior for the layer below. In order to perform scalable inference for big corpora, both stochastic gradient Markov chain Monte Carlo (SG-MCMC) [19]–[22] and stochastic variational inference (SVI) [9], [12], [23], [24] have been developed for topic models. Despite being able to infer a multilayer representation of a text corpus, they usually rely on an iterative procedure to infer the latent representation of a new document at the testing stage, regardless of whether variational inference or MCMC

is used. The potential need of a large number of iterations per testing document makes them unattractive when real-time processing is desired. For example, one may need to rapidly extract the topic-proportion vector of a document and use it for downstream analysis, such as identifying key topics and retrieving related documents. In addition, they often make the restrictive mean-field assumption [24], require sophisticated variance reduction techniques [12], and use a single learning rate for different variables across all layers [21], [25], [26], making it difficult to generalize them to deep probabilistic models.

A potential solution is to construct a variational autoencoder (VAE) that learns the parameters of an inference network (recognition model or encoder) jointly with those of the generative model (decoder) [27], [28]. However, most existing VAEs rely on Gaussian latent variables, with the neural networks (NNs) acting as nonlinear connections between adjacent layers [29]–[31]. A primary reason is that there is a simple reparameterization trick for Gaussian latent variables that allows efficiently computing the noisy gradients of the evidence lower bound (ELBO) with respect to the NN parameters. Unfortunately, Gaussian based distributions often fail to well approximate the posterior distributions of sparse, nonnegative, and skewed document latent representations. For example, Srivastava et al. [32] propose autoencoding variational inference for topic models (AVITM), as shown in Fig. 1b, which utilizes the logistic-normal distribution to approximate the posterior distribution of the latent representation of a document; even though the generative model is LDA [1], a basic single-hidden-layer topic model, due to the insufficient ability of the logistic-normal distribution to model sparsity, AVITM has to rely on some heuristics to force the latent representation of a document to be sparse. To overcome this limitation, Knowles [33] introduces a reparameterization method for the gamma distribution that relies on inefficient numerical approximation; Ruiz et al. [34] develop generalized reparameterization (Grep) to extend the reparameterization gradient to a wider class of variational distributions utilizing invertible transformations, leading to transformed distributions that only weakly depend on the variational parameters; and Naesseth et al. [35] further improve Grep with rejection sampling variational inference (RSVI) that achieves lower variance and faster speed via a rejection sampling algorithm at the cost of introducing more random noisy. Another common shortcoming of existing VAEs is that they often only provide a point estimate for the global parameters of the generative model, and hence their inference network is optimized to approximate the posterior of the local parameters conditioning on the data and that point estimate, rather than a full posterior, of the global parameters. In addition, from a probabilistic modeling point of view, the VAE inference network is often merely a shallow probabilistic model, whose parameters are deterministically nonlinearly transformed from the observations via a non-probabilistic deep NN.

B. Motivations and contributions

To address the aforementioned constrains of existing topic models and move beyond Gaussian latent variable based deep generative models and inference, we develop deep autoencoding topic model (DATM). DATM uses a deep topic model as its decoder and a deterministic-upward–stochastic-downward network as its encoder, and jointly trains them with a hybrid Bayesian inference, integrating both SG-MCMC [21], [22], [36] and a multilayer Weibull distribution based inference network. The distinctions of DATM are summarized as follows.

- DATM is related to a usual VAE in having both a decoder and encoder, but differs from it in a number of ways: 1) Deep latent Dirichlet allocation (DLDA), a probabilistic deep topic model equipped with a gamma belief network, acts as the generative model; 2) Inspired by the upward-downward Gibbs sampler of DLDA, as sketched in Fig. 1c, the inference network of DATM uses an upward-downward structure, as shown in Fig. 1a, to combine a non-probabilistic bottom-up deep NN and a probabilistic top-down deep generative model, with the ℓth hidden layer of the generative model linked to both the $(\ell+1)$th hidden layer of itself and the ℓth hidden layer of the deep NN; 3) A hybrid of SG-MCMC and autoencoding variational inference is employed to infer both the posterior distribution of the global parameters, represented as collected posterior MCMC samples, and a VAE that approximates the posterior distribution of the local parameters given the data and a posterior sample (rather than a point estimate) of the global parameters; 4) We use the Weibull distributions in the inference network to approximate gamma distributed conditional posteriors, exploiting the facts that the Weibull and gamma distributions have similar probability density functions (PDFs), the Kullback–Leibler (KL) divergence from the Weibull to gamma distributions is analytic, and a Weibull random variable can be reparameterized with uniform random noise.
- In probabilistic topic models, the document-specific topic proportions are commonly used as features for downstream analysis such as document classification. Although the unsupervisedly extracted features can be used to train a classifier [1], [17], [18], they provide relatively poor discrimination power [37]. Although some discriminative models [5], [38], [39] are integrated into LDA to improve its discrimination power, their single-hidden-layer structure clearly limits their ultimate potential for satisfactorily representing high-dimensional and sparse document data. Exploiting the multi-layer structure of DATM, we further propose a supervised deep topic model, referred to as supervised DATM (sDATM), that combines the flexibility of DATM in describing the documents and the discriminative power of deep NNs under a principled probabilistic framework. Distinct from supervised LDA and its extensions [5], [38], [39], the features at different layers of sDATM exhibit different statistical properties, and hence are combined together to boost their discriminative power.
- DATM provides interpretable hierarchical topics, which vary from very specific to increasingly more general when moving towards deeper layers. In DATM, the number of topics in a layer, i.e., the width of that layer, is automatically learned from the data given a fixed budget on the width of the first layer, with the help of the gamma-negative binomial process and a greedy layer-wise training strategy [4].

The remainder of the paper is organized as follows. Section II

introduces a deep probabilistic autoencoder for topic modeling and develops a hybrid Bayesian inference algorithm to perform efficient scalable inference. With label information, a supervised deep topic model is introduced in Section III. Section IV reports a series of experiments on document representation and classification to evaluate the proposed models. Section V concludes the paper. We note that parts of the work presented here first appeared in Cong et al. [22] and Zhang et al. [40]. In this paper, we unify related materials in both conference publications and provide expansion to supervised deep topic modeling. Furthermore, to infer from the data rather than predetermining the network structure, given a fixed budget on the width of the first layer, we combine Bayesian nonparametrics and greedy layer-wise training for DATM to learn the width of each added hidden layer, with all the added hidden layers jointly trained, leading to further improved performance.

II. Deep Autoencoding Topic Model

In what follows, we propose DATM that uses a deep hierarchical Bayesian model as the generative model (decoder), and a deterministic-upward–stochastic-downward network as the recognition model (encoder, inference network).

A. Document decoder: deep latent Dirichlet allocation

To capture a hierarchical document latent representation, DATM uses PGBN [17], a deep probabilistic topic model, as the decoder. Choosing a deep generative model as its decoder distinguishes DATM from AVITM [32], which uses a "shallow" LDA as its decoder, and from the conventional VAE, which often employs a "shallow" (transformed) Gaussian distribution as its decoder with parameters deterministically and nonlinearly transformed from the observation via "black-box" deep NNs.

To model high-dimensional multivariate sparse count vectors $\boldsymbol{x}_n \in \mathbb{Z}^{K_0}$, where $\mathbb{Z} = \{0, 1, \ldots\}$, under the Poisson likelihood, the PGBN generative model with L hidden layers, from top to bottom, can be expressed as

$$\begin{aligned}
&\boldsymbol{\theta}_n^{(L)} \sim \text{Gam}\left(\boldsymbol{r}, 1/c_n^{(L+1)}\right), \boldsymbol{r} \sim \text{Gam}(\gamma_0/K_L, 1/c_0)\\
&\boldsymbol{\theta}_n^{(l)} \sim \text{Gam}\left(\boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, 1/c_n^{(l+1)}\right), \quad l = 1, \cdots, L-1\\
&\boldsymbol{x}_n \sim \text{Pois}\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right),
\end{aligned} \tag{1}$$

where the hidden units (topic weights) $\boldsymbol{\theta}_n^{(l)} \in \mathbb{R}_+^{K_l}$ of layer l are factorized into the product of the factor loading $\boldsymbol{\Phi}^{(l+1)} \in \mathbb{R}_+^{K_l \times K_{l+1}}$ and hidden units of layer $l+1$ under the gamma distribution. For scale identifiability and ease of inference and interpretation, PGBN further places a simplex constraint on each column of $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^L$ via a Dirichlet prior as $\boldsymbol{\Phi}_k^{(l)} \sim \text{Dir}\left(\eta^{(l)}\mathbf{I}_{K_{l-1}}\right)$, where $\mathbf{I}_{K_{l-1}}$ is a vector of K_{l-1} ones. The gamma shape parameters $\boldsymbol{r} = (r_1, \cdots, r_{K_L})^T$ at the top layer are shared across all $\boldsymbol{x}_n$ and $\{1/c_n^{(l)}\}_{l=2}^{L+1}$ are gamma scale parameters.

Using the law of total expectation, we have

$$\mathbb{E}\left[\boldsymbol{x}_n \,\middle|\, \boldsymbol{\theta}_n^{(l)}, \left\{\boldsymbol{\Phi}^{(t)}, c_n^{(t)}\right\}_{t=1}^{l}\right] = \left[\prod_{t=1}^{l}\boldsymbol{\Phi}^{(t)}\right]\frac{\boldsymbol{\theta}_n^{(l)}}{\prod_{t=2}^{l} c_n^{(t)}}, \tag{2}$$

which means the conditional expectation of $\boldsymbol{x}_n$ on layer l is a linear combination of the columns in $\prod_{t=1}^{l}\boldsymbol{\Phi}^{(t)}$, with $\boldsymbol{\theta}_n^{(l)}$ viewed as a document-dependent topic-weight vector that can be used for downstream analysis, such as document classification and retrieval. Furthermore, $\prod_{t=1}^{l-1}\boldsymbol{\Phi}^{(t)}\phi_k^{(l)}$ can be viewed as the projection of topic $\phi_k^{(l)}$ to the bottom data layer, which can be used to visualize the topics at different layers. An example of the hierarchical topic structure is illustrated in Fig. 5. The inferred topics of this model tend to be very specific at the bottom layer and become increasingly more general when moving upwards (deeper).

Denote $q_n^{(l+1)} = \log(1+q_n^{(l)}/c_n^{(l+1)})$ for $l = 1, \cdots, L$, where $q_n^{(1)} = 1$, $p_j^{(l)} = 1 - e^{-q_j^{(l)}}$, and $m \sim \text{SumLog}(x, p)$ as the sum-logarithmic distribution [41]. With all the gamma distributed hidden units marginalized out, PGBN can also be represented as deep LDA (DLDA) [22], expressed as

$$\begin{aligned}
&x_{kn}^{(L+1)} \sim \text{Pois}(r_k q_n^{(L+1)}),\ k = 1, \ldots, K_L,\\
&m_{kn}^{(L)(L+1)} \sim \text{SumLog}(x_{kn}^{(L+1)}, p_n^{(L+1)}),\ k = 1, \ldots, K_L,\\
&\qquad\qquad \cdots\\
&\left(x_{vkn}^{(l)}\right)_{v=1,K_{l-1}} \sim \text{Mult}\left(m_{kn}^{(l)(l+1)}, \boldsymbol{\phi}_k^{(l)}\right),\ k = 1, \ldots, K_l,\\
&x_{kn}^{(l)} = \sum_{k'=1}^{K_l} x_{kk'n}^{(l)}, \quad k = 1, \ldots, K_{l-1},\\
&m_{kn}^{(l-1)(l)} \sim \text{SumLog}\left(x_{kn}^{(l)}, p_j^{(l)}\right), \quad k = 1, \ldots, K_{l-1},\\
&\qquad\qquad \cdots\\
&\left(x_{vkn}^{(1)}\right)_{v=1,K_0} \sim \text{Mult}\left(m_{kn}^{(1)(2)}, \boldsymbol{\phi}_k^{(1)}\right),\ k = 1, \ldots, K_1\\
&x_{kn}^{(1)} = \sum_{k'=1}^{K_1} x_{kk'n}^{(1)},\ k = 1, \ldots, K_0.
\end{aligned} \tag{3}$$

For simplicity, below we use DLDA to refer to both the PGBN and DLDA representations of the same underlying deep generative model. Note the single-hidden-layer version of DLDA reduces to Poisson factor analysis [14], which is closely related to LDA. To make DLDA be scalable to big corpora, in the following, we develop a SG-MCMC based algorithm.

SG-MCMC. For a statistical model with likelihood $p(\boldsymbol{x} \mid \boldsymbol{z})$ and prior $p(\boldsymbol{z})$, where $\boldsymbol{z}$ denotes the set of all global variables, one may follow the general framework for SG-MCMC [19] to express the sampling equation as

$$\begin{aligned}
\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \epsilon_t &\left\{-\left[\mathbf{D}(\boldsymbol{z}_t) + \mathbf{Q}(\boldsymbol{z}_t)\right]\nabla H(\boldsymbol{z}_t) + \Gamma(\boldsymbol{z}_t)\right\}\\
&+ \mathcal{N}\left(\mathbf{0}, \epsilon_t\left[2\mathbf{D}(\boldsymbol{z}_t) - \epsilon_t\mathbf{B}_t\right]\right),
\end{aligned} \tag{4}$$

where ϵ_t denotes the step size at step t, $H(\boldsymbol{z}) = -\ln p(\boldsymbol{z}) - \rho\Sigma_{\boldsymbol{x}\in\hat{\mathbf{X}}}\ln p(\boldsymbol{x}|\boldsymbol{z})$, $\Gamma_i(\boldsymbol{z}_t) = \sum_j \frac{\partial}{\partial z_{jt}}[\mathbf{D}_{ij}(\boldsymbol{z}_t) + \mathbf{Q}_{ij}(\boldsymbol{z}_t)]$, $\hat{\mathbf{X}}$ the mini-batch, ρ the ratio of the dataset size $|\mathbf{X}|$ to the mini-batch size $|\hat{\mathbf{X}}|$, and $\mathbf{B}_t$ an estimate of the stochastic gradient noise variance satisfying a positive definite constraint as $2\mathbf{D}(\boldsymbol{z}_t) - \epsilon_t\mathbf{B}_t \succ \mathbf{0}$. Under this framework, stochastic gradient Riemannian Langevin dynamics (SGRLD) [20] is proposed for LDA, with $\mathbf{D}(\boldsymbol{z}) = \mathbf{G}(\boldsymbol{z})^{-1}$, $\mathbf{Q}(\boldsymbol{z}) = \mathbf{0}$, and $\mathbf{B}_t = \mathbf{0}$, where

$$\mathbf{G}(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{\Pi}|\boldsymbol{z}}\left[-\frac{\partial^2}{\partial \boldsymbol{z}^2}\ln p(\boldsymbol{\Pi}|\boldsymbol{z})\right] \tag{5}$$

is the Fisher information matrix (FIM) that is widely used to precondition the gradients to adjust the learning rates, where $\mathbf{\Pi}$ denotes the set of all observed and local variables. In general, it is difficult to calculate the FIM. Fortunately, DLDA admits a block-diagonal FIM that is easy to work with.

Fisher information matrix. Since the topic $\boldsymbol{\phi}_k^{(l)} \in \mathbb{R}_+^{K_{l-1}}$ is a vector that lies on the probabilistic simplex, we use $\boldsymbol{\phi}_k^{(l)} = \left(\varphi_{1k}^{(l)}, \cdots, \varphi_{(K_{l-1}-1)k}^{(l)}, 1-\Sigma_{v<K_{l-1}}\varphi_{vk}^{(l)}\right)^T$ as the reduced-mean parameterization of $\boldsymbol{\phi}_k^{(l)}$, which is different from the expanded-mean parameterization used by SGRLD [20]. Under the DLDA representation shown in (3), the likelihood is fully factorized with respect to the global parameters $\boldsymbol{z} = \{\boldsymbol{\varphi}_1^{(1)}, \ldots, \boldsymbol{\varphi}_{K_L}^{(L)}, \boldsymbol{r}\}$, leading to a FIM that admits a block diagonal form as

$$\mathbf{G}(\boldsymbol{z}) = \text{diag}\left[\mathbf{I}\left(\boldsymbol{\varphi}_1^{(1)}\right), \cdots, \mathbf{I}\left(\boldsymbol{\varphi}_{K_L}^{(L)}\right), \mathbf{I}\left(\boldsymbol{r}\right)\right], \quad (6)$$

$$\mathbf{I}\left(\boldsymbol{\varphi}_k^{(l)}\right) = M_k^{(l)}\left[\text{diag}\left(1/\boldsymbol{\varphi}_k^{(l)}\right) + \mathbf{1}\mathbf{1}^T/(1-\boldsymbol{\varphi}_{\cdot k}^{(l)})\right], \quad (7)$$

$$\mathbf{I}\left(\boldsymbol{r}\right) = M^{(L+1)}\text{diag}\left(r_1^{-1}, \ldots, r_{K_L}^{-1}\right), \quad (8)$$

where the symbol "$\cdot$" denotes summing over the corresponding index, $M_l^{(l)} = \mathbb{E}\left[m_{k\cdot}^{(l)(l+1)}\right] = \mathbb{E}\left[x_{\cdot k\cdot}^{(l)}\right]$, and $M^{(L+1)} = \mathbb{E}\left[q_{\cdot}^{(L+1)}\right]$. Note that the block diagonal structure of the FIM for DLDA makes it computationally appealing to apply its inverse for preconditioning. To utilize (4) to develop SG-MCMC for DLDA under the reduced-mean parameterization, similar to SGRLD, we let $\mathbf{D}(\boldsymbol{z}) = \mathbf{G}(\boldsymbol{z})^{-1}$, $\mathbf{Q}(\boldsymbol{z}) = \mathbf{0}$, and $\mathbf{B}_t = \mathbf{0}$. Relying on the FIM to automatically adjust relative learning rates for different parameters across all layers and topics, we only need to choose a single step size ε_t for all. Moreover, the block-diagnoal structure of FIM will be carried over to its inverse $\mathbf{D}(\boldsymbol{z})$, leading to a computationally efficient way to perform updating by (4), as described below.

Inference on the probability simplex. Using the DLDA representation in (3) and reduced-mean parameterization of simplex-constrained vectors, we derive a block-diagonal FIM as in (39). Besides this advantage, we describe another reason for our choice in the following discussion, where we ignore the layer-index superscript $^{(l)}$ for brevity and assume $\boldsymbol{\phi}_k = (\boldsymbol{\varphi}_k^T, 1-\varphi_{\cdot k})^T \in \mathbb{R}_+^V$ lies on a V-dimensional simplex.

With (3) and the Dirichlet-multinomial conjugacy, taking the gradient with respect to $\boldsymbol{\varphi}_k$ of the summation of the log-likelihood of a mini-batch $\tilde{X}$ scaled by $\rho = |X|/|\tilde{X}|$ and the logarithm of the Dirichlet prior, we have

$$\nabla_{\boldsymbol{\varphi}_k}\left[-H(\boldsymbol{\varphi}_k)\right] = \frac{\rho\bar{\boldsymbol{x}}_{:k\cdot} + \eta - 1}{\boldsymbol{\varphi}_k} - \frac{\rho\tilde{x}_{vk\cdot} + \eta - 1}{1-\varphi_{\cdot k}}, \quad (9)$$

where $\tilde{x}_{vk\cdot} = \sum_{n:x_n\in\tilde{X}} x_{vkn}$, $\bar{\boldsymbol{x}}_{:k\cdot} = \left(\tilde{x}_{1k\cdot}, \cdots, \tilde{x}_{(V-1)k\cdot}\right)^T$. Note the gradient in (9) becomes unstable when some components of $\boldsymbol{\varphi}_k$ approach zeros, a key reason that this approach is mentioned but considered as an unsound choice in Patterson & Teh [20]. However, after preconditioning the noisy gradient in (9) with the inverse of the FIM, it is intriguing to find out that the stability issue completely disappears. More specifically, by substituting (39), (9), and $\Gamma(\boldsymbol{\varphi}_k)$, whose derivation is given in the Supplement, into the SG-MCMC update equation (4), the sampling of $\boldsymbol{\varphi}_k$ becomes

$$(\boldsymbol{\varphi}_k)_{t+1} = \Big[(\boldsymbol{\varphi}_k)_t + \frac{\varepsilon_t}{M_k}[(\rho\tilde{\boldsymbol{x}}_{:k\cdot}+\eta) - (\rho\tilde{x}_{\cdot k\cdot}+\eta V)(\boldsymbol{\varphi}_k)_t] + \mathcal{N}\left(0, \frac{2\varepsilon_t}{M_k}\left[\text{diag}(\boldsymbol{\varphi}_k)_t - (\boldsymbol{\varphi}_k)_t(\boldsymbol{\varphi}_k)_t^T\right]\right)\Big]_{\triangle}, \quad (10)$$

where $[\cdot]_{\triangle}$ represents the constraint that $\varphi_{vk} \geq 0$ and $\varphi_{\cdot k} = \sum_{v=1}^{V-1}\varphi_{vk} \leq 1$.

Note while the stability issue has now been solved, naively sampling from the multivariate normal (MVN) distribution in (10), even without the $[\cdot]_{\triangle}$ constraint, is computationally expensive as a Cholesky decomposition of the non-diagonal covariance matrix has $O((V-1)^3)$ complexity [42]. Fortunately, following Theorem 2 in Cong et al. [43], we may equivalently draw $\boldsymbol{\phi}_k$ from a V-dimensional MVN that has a diagonal covariance matrix and is subject to the simplex constraint, with $O(V)$ complexity, as

$$(\boldsymbol{\phi}_k)_{t+1} = \Big[(\boldsymbol{\phi}_k)_t + \frac{\varepsilon_t}{M_k}[(\rho\tilde{\boldsymbol{x}}_{:k\cdot}+\eta) - (\rho\tilde{x}_{\cdot k\cdot}+\eta V)(\boldsymbol{\phi}_k)_t] + \mathcal{N}\left(0, \frac{2\varepsilon_t}{M_k}\text{diag}\left(\boldsymbol{\phi}_k\right)_t\right)\Big]_{\angle}, \quad (11)$$

where $[\cdot]_{\angle}$ denotes a simplex constraint that $\phi_{vk} \geq 0$ and $\Sigma_{v=1}^V\phi_{vk} = 1$. More details about (10) and (11) can be found in Examples 1-3 of Cong et al. [43].

Similarly, with the gamma-Poisson conjugacy for $\boldsymbol{r}$, we have $\Gamma_k(\boldsymbol{r}) = 1/M^{(L+1)}$, whose detailed derivation is deferred to the Supplement, and hence the update of $\boldsymbol{r}$ as

$$\boldsymbol{r}_{t+1} = \left|\boldsymbol{r}_t + \frac{\varepsilon_t}{M^{(L+1)}}\left[\left(\rho\tilde{\boldsymbol{x}}_{:\cdot}^{(L+1)} + \frac{\gamma_0}{K_L}\right) - \boldsymbol{r}_t\left(c_0 + \rho\tilde{q}_{\cdot}^{(L+1)}\right)\right] + \mathcal{N}\left(\mathbf{0}, \frac{2\varepsilon_t}{M^{(L+1)}}\text{diag}(\boldsymbol{r}_t)\right)\right|. \quad (12)$$

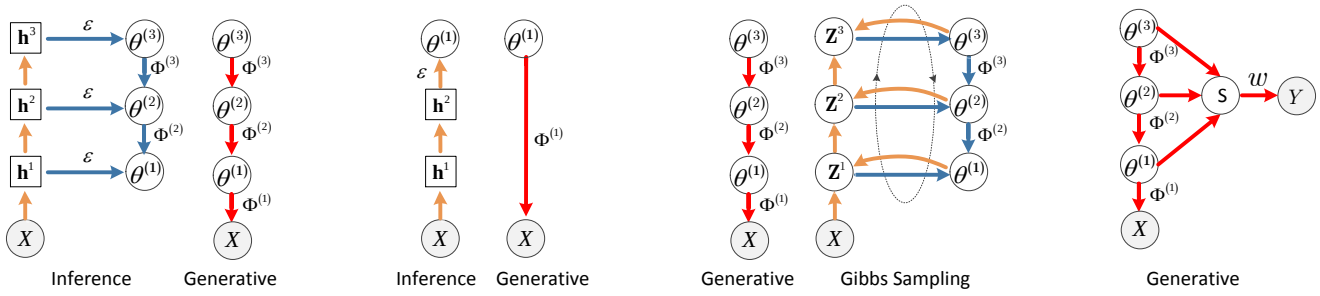
Topic-layer-adaptive step-size. Note that $\left\{M_k^{(l)}\right\}_{l=1}^L$ and $M^{(L+1)}$ appearing in (11) and (12) are expectations over all local variables that need to be approximately calculated. We update them using annealed weighting [44] as

$$M_k^{(l)} = \left(1-\varepsilon_t^{'}\right)M_k^{(l)} + \varepsilon_t^{'}\rho\mathbb{E}\left[x_{\cdot k\cdot}^{(l)}\right], \quad (13)$$

$$M_k^{(L+1)} = \left(1-\varepsilon_t^{'}\right)M_k^{(L+1)} + \varepsilon_t^{'}\rho\mathbb{E}\left[\tilde{q}_{\cdot}^{(L+1)}\right], \quad (14)$$

where the expectation $\mathbb{E}[\cdot]$ denotes averaging over the collected MCMC samples. In DATM to be discussed below, this can be approximated by one sample from the introduced document encoder (22), instead of collected MCMC samples. For simplicity, we set $\varepsilon_t^{'} = \varepsilon_t$, which is found to work well in practice. Note although it appears that a common step-size ε_t is set for all layers, the effective step-sizes are $\varepsilon_t/M_k^{(l)}$, which differ for all layers ($l \in \{1, \cdot, \cdot, \cdot L\}$) and topics ($k \in \{1, \cdot, \cdot, \cdot K_l\}$). For this reason, We refer to the proposed SG-MCMC as topic-layer-adaptive stochastic gradient Riemannian (TLASGR) MCMC.

Despite having attractive properties and scalable inference via TLASGR-MCMC, the power of DLDA is limited in that it has to take a potentially large number of MCMC iterations



(a) DATM (b) AVITM (c) DLDA (d) sDPATM

Fig. 1: (a-b): Inference (or encoder/recognition) and generative (or decoder) models for (a) DATM and (b) AVITM; (c) the generative model and a sketch of the upward-downward Gibbs sampler of DLDA, where $\mathbf{Z}^l$ are augmented latent counts that are upward sampled in each Gibbs sampling iteration. Circles are stochastic variables and squares are deterministic variables. The orange and blue arrows denote the upward and downward information propagation respectively, and the red ones denote the data generation; (d) the generative model of sDPATM.

to infer the latent representation of a test observation, and it is difficult to utilize available side information such as class labels. To address these issues, in the following we first develop a deep document encoder network.

B. Document encoder: Weibull upward-downward variational encoder

To perform fast inference for out-of-sample predictions, we are motivated to construct an inference network that maps the observations directly to the posterior distributions of their latent representations and hence avoid performing any iterative updates at the test time. Variational auto-encoder (VAE) [27], [28] becomes an idea candidate for this purpose. However, its success so far is mostly restricted to Gaussian distributed latent variables, and does not generalize well to model sparse, nonnegative, and skewed latent document representations. To this end, below we propose Weibull upward-downward variational encoder (WUDVE) to efficiently produce a document's multi-layer latent representation under DLDA.

To maximize the marginal likelihood $p(\boldsymbol{x})$ under DLDA, one may choose a usual strategy of variational Bayes [45] to maximize the ELBO of $p(\boldsymbol{x})$ that can be expressed as

$$L = \sum_{n=1}^{N} \mathbb{E}\left[\ln p\left(\boldsymbol{x}_n \,|\, \boldsymbol{\Phi}^{(1)}, \boldsymbol{\theta}_n^{(1)}\right)\right] - \sum_{n=1}^{N}\sum_{l=1}^{L} \mathbb{E}\left[\ln \frac{q\left(\boldsymbol{\theta}_n^{(l)} \,|\, \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{\theta}_n^{(l+1)}\right)}{p\left(\boldsymbol{\theta}_n^{(l)} \,|\, \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{\theta}_n^{(l+1)}\right)}\right], \quad (15)$$

where $\boldsymbol{\Phi}^{(L+1)} := \boldsymbol{r}$, $\boldsymbol{\theta}_n^{(L+1)} := \emptyset$, and the expectations are taken with respect to the variational distribution as

$$q\left(\{\boldsymbol{\theta}_n^{(l)}\}_{n=1,l=1}^{N,L}\right) = \prod_{n=1}^{N}\prod_{l=1}^{L} q\left(\boldsymbol{\theta}_n^{(l)} \,|\, \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{\theta}_n^{(l+1)}\right). \quad (16)$$

To simplify the optimization, one often resorts to the mean-field assumption that factorizes the variational distribution as

$$q\left(\{\boldsymbol{\theta}_n^{(l)}\}_{n=1,l=1}^{N,L}\right) = \prod_{n=1}^{N}\prod_{l=1}^{L} q\left(\boldsymbol{\theta}_n^{(l)}\right). \quad (17)$$

Furthermore, to achieve fast out-of-sample prediction with autoencoding variational inference, one may consider a gamma distribution based inference network as $q(\boldsymbol{\theta}_n^{(l)} \,|\, \boldsymbol{x}_n) = \text{Gamma}(f_{\mathbf{W}}^{(l)}(\boldsymbol{x}_n), g_{\mathbf{W}}^{(l)}(\boldsymbol{x}_n))$ to model sparse and nonnegative latent document representation, where $f^{(l)}$ and $g^{(l)}$ are related DNNs parameterized by $\mathbf{W}$. However, it is hard to efficiently compute the gradient of the ELBO with respect to $\mathbf{W}$, especially if $L \geq 2$, due to the difficulty to reparameterize a gamma random variable [27], [33], [34], motivating us to identify a surrogate distribution that can not only well approximate the gamma distribution, but also be easily reparameterized. Below we show the Weibull distribution is an ideal choice.

Weibull variational posterior. A main reason that we choose the Weibull distribution to construct the inference network is that the Weibull and gamma distributions have similar PDFs, which makes it possible to model sparse and nonnegative latent representation:

$$\begin{aligned} &\text{Weibull PDF: } P(x \,|\, k, \lambda) = \frac{k}{\lambda^k} x^{k-1} e^{(-x/\lambda)^k}, \\ &\text{Gamma PDF: } P(x \,|\, \alpha, 1/\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \end{aligned} \quad (18)$$

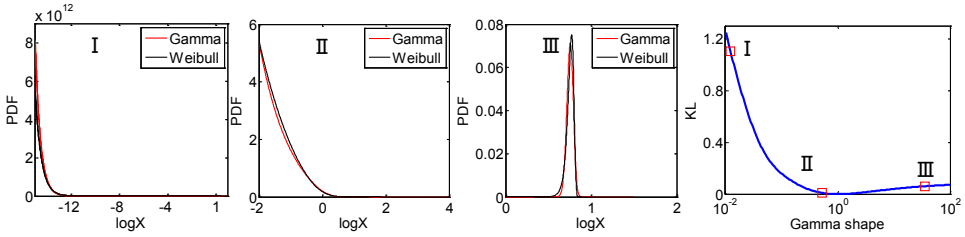
where $x \in \mathbb{R}_+$. Another reason is due to a simple reparameterization for $x \sim \text{Weibull}(k, \lambda)$ as

$$x = \lambda(-\ln(1-\epsilon))^{1/k}, \ \epsilon \sim \text{Uniform}(0, 1), \quad (19)$$

leading an easy-to-compute gradient when maximizing the ELBO. Moreover, denoting γ as the Euler–Mascheroni constant, the KL-divergence from the gamma to Weibull distribution has an analytic expression as

$$\begin{aligned} &\text{KL}(\text{Weibull}(k, \lambda) || \text{Gamma}(\alpha, 1/\beta)) = -\alpha \ln \lambda + \frac{\gamma\alpha}{k} + \ln k \\ &+ \beta\lambda\Gamma\left(1 + \frac{1}{k}\right) - \gamma - 1 - \alpha \ln \beta + \ln \Gamma(\alpha), \end{aligned} \quad (20)$$

which helps reduce the variance when evaluating the gradient of the ELBO [27]. Minimizing this KL divergence, one can identify the two parameters of a Weibull distribution to approximate a given gamma one. As shown in Fig. 2, the inferred Weibull distribution in general accurately approximates



(a) KL = 0.96 (b) KL = 0.01 (c) KL = 0.06 (d) KL change

Fig. 2: The KL divergence from the inferred Weibull distribution to the target gamma one as (a) Gamma$(0.05, 1)$, (b) Gamma$(0.5, 1)$, and (c) Gamma$(5, 1)$. Subplot (d) shows the KL divergence as a function of the gamma shape parameter, where the gamma scale parameter is fixed at 1.

the target gamma one, as long as the gamma shape parameter is neither too close to zero nor too large.

Upward-downward information propagation. With the DLDA upward-downward Gibbs sampler sketched in Fig. 1c and the corresponding sampling equation

$$(\boldsymbol{\theta}_n^{(l)} \mid -) \sim \text{Gamma}\left(\boldsymbol{m}_n^{(l)(l+1)} + \boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, f(p_n^{(l)}, c_n^{(l+1)})\right), \tag{21}$$

where $\boldsymbol{m}_n^{(l)(l+1)}$ and $p_n^{(l)}$ are latent random variables constituted by information upward propagated to layer l, it is clear that the conditional posterior of $\boldsymbol{\theta}_n^{(l)}$ is related to both the information at the higher (prior) layer, and that upward propagated to the current layer via a series of data augmentation and marginalization steps; see Zhou et al. [18] for more details. Considering that VAE-like models usually build the upward propagation but ignore the impact of the prior, inspired by the instructive upward-downward information propagation in Gibbs sampling, as shown in Fig. 1a, we construct WUDVE, the inference network of our model, as $q(\boldsymbol{\theta}_n^{(L)} \mid \boldsymbol{h}_n^{(L)}) \prod_{l=1}^{L-1} q(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{h}_n^{(l)}, \boldsymbol{\theta}_n^{(l+1)})$, where

$$\begin{aligned} &q(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{h}_n^{(l)}, \boldsymbol{\theta}_n^{(l+1)}) \\ &= \text{Weibull}(\boldsymbol{k}_n^{(l)} + \boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, \boldsymbol{\lambda}_n^{(l)}). \end{aligned} \tag{22}$$

The Weibull distribution is used to approximate the gamma distributed conditional posterior, and its parameters $\boldsymbol{k}_n^{(l)} \in \mathbb{R}^{K_l}$ and $\boldsymbol{\lambda}_n^{(l)} \in \mathbb{R}^{K_l}$ are both deterministically transformed from the observation $\boldsymbol{x}_n$ using the NNs, as illustrated in Fig. 1a and specified as

$$\boldsymbol{k}_n^{(l)} = \ln[1 + \exp(\mathbf{W}_1^{(l)}\boldsymbol{h}_n^{(l)} + \boldsymbol{b}_1^{(l)})], \tag{23}$$

$$\boldsymbol{\lambda}_n^{(l)} = \ln[1 + \exp(\mathbf{W}_2^{(l)}\boldsymbol{h}_n^{(l)} + \boldsymbol{b}_2^{(l)})], \tag{24}$$

$$\boldsymbol{h}_n^{(l)} = \ln[1 + \exp(\mathbf{W}_3^{(l)}\boldsymbol{h}_n^{(l-1)} + \boldsymbol{b}_3^{(l)})], \tag{25}$$

where $\boldsymbol{h}_n^{(0)} = \log(1 + \boldsymbol{x}_n)$, $\mathbf{W}_1^{(l)} \in \mathbb{R}^{K_l \times K_l}$, $\mathbf{W}_2^{(l)} \in \mathbb{R}^{K_l \times K_l}$, $\mathbf{W}_3^{(l)} \in \mathbb{R}^{K_l \times K_{l-1}}$, $\boldsymbol{b}_1^{(l)} \in \mathbb{R}^{K_l}$, $\boldsymbol{b}_2^{(l)} \in \mathbb{R}^{K_l}$, and $\boldsymbol{b}_3^{(l)} \in \mathbb{R}^{K_l}$. This upward-downward inference network is distinct from that of a usual VAE, where it is common that the inference network has a pure bottom-up structure and only interacts with the generative model via the ELBO [27], [31]. Note that it does not follow mean-field variational Bayes to make a fully factorized assumption as in (17).

Comparing Figs. 1c and 1a shows that in each iteration, both Gibbs sampling in DLDA and the hybrid Bayesian inference in DATM have not only upward information propagations (orange arrows), but also downward ones (blue arrows), but there are distinctions between their underlying implementations. Gibbs sampling in Fig. 1c does not have an inference network and needs the local variables $\boldsymbol{\theta}_n^{(l)}$ to help perform stochastic upward information propagation, whereas DATM in Fig. 1a uses a ladder network to combine a deterministic upward and stochastic downward information propagation, without relying on the local variables $\boldsymbol{\theta}_n^{(l)}$. It is also interesting to notice that the upward-downward structure, motivated by the upward-downward Gibbs sampler of DLDA, is closely related to the ladder structure used in the ladder VAE [29]. However, to combine the bottom-up and top-down information, ladder VAE relies on some heuristics restricted to Gaussian latent variables.

C. Weibull hybrid autoencoding inference (WHAI)

Based on the above discussion, in DATM, we need to infer the topic parameters $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$ of the decoder network and the NN parameters $\boldsymbol{\Omega} = \{\mathbf{W}_1^{(l)}, \boldsymbol{b}_1^{(l)}, \mathbf{W}_2^{(l)}, \boldsymbol{b}_2^{(l)}, \mathbf{W}_3^{(l)}, \boldsymbol{b}_3^{(l)}\}_{1,L}$ of the encoder network.

Rather than merely finding point estimates, we describe in Algorithm 1 how to combine TLASGR-MCMC and WUDVE into a hybrid SG-MCMC/VAE inference algorithm, which infers posterior samples for $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$ and $\boldsymbol{\Omega}$. An important step of Algorithm 1 is calculating the gradient of the ELBO in (15) with respect to the NN parameters $\boldsymbol{\Omega}$, which is important to the success of a variational inference algorithm [9], [12], [27], [28], [34], [46]. Thanks to the choice of the Weibull distribution, the second term of the ELBO in (15) is analytic, and due to simple reparameterization of the Weibull distribution, the gradient of the first term of the ELBO with respect to $\boldsymbol{\Omega}$ can be accurately evaluated, achieving satisfactory performance using as few as a single Monte Carlo sample, as shown in our experimental results. Thanks to the architecture of DATM using the inference network, for a new mini-batch, different from Cong et al. [22] that run hundreds of MCMC iterations to collect posterior samples for local variables, we can directly find the conditional posterior of $\{\boldsymbol{\theta}_n^{(l)}\}_{l=1}^{L}$ given $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$ and the stochastically updated $\boldsymbol{\Omega}$, with which we can sample the local parameters and then use TLASGR-MCMC to stochastically update the global parameters $\{\boldsymbol{\Phi}^{(l)}\}_{1,L}$.

D. Learning the network structure with layer-wise training

Distinct from some existing unsupervised learning algorithms that train deep networks in a greedy layer-wise manner, such

Algorithm 1 Hybrid stochastic-gradient MCMC and autoencoding variational inference for DATM

Input: Observed data $\{\boldsymbol{x}_n\}_n$, the structure of DATM, and hyper-parameters.
Output: Global parameters of DATM $\{\boldsymbol{\Phi}^{(l)}\}_{1,L}$ and $\boldsymbol{\Omega}$.
Set mini-batch size m;
Initialize encoder parameters $\boldsymbol{\Omega}$ and decoder parameters $\{\boldsymbol{\Phi}^{(l)}\}_{1,L}$.
for $iter = 1, 2, \cdots$ **do**
  Randomly select a mini-batch of m documents to form a subset $\mathbf{X} = \{\boldsymbol{x}_i\}_{1,m}$;
  Draw random noise $\{\varepsilon_i^l\}_{i=1,l=1}^{m,L}$ from uniform distribution;
  Calculate $\nabla_{\boldsymbol{\Omega}} L\left(\boldsymbol{\Omega}, \boldsymbol{\Phi}^{\{l\}}; \mathbf{X}, \varepsilon_i^l\right)$ according to (15), and update $\boldsymbol{\Omega}$;
  Sample $\left\{\boldsymbol{\theta}_i^{(l)}\right\}_{i=1,l=1}^{m,L}$ from (22) via $\boldsymbol{\Omega}$;
  for $l = 1, \cdots, L+1$ and $k = 1, \cdots, K_l$ **do**
    Update $M_k^{(l)}$ with (13); then topics $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$ with (11) and $\boldsymbol{r}$ with (12).
  end for
end for

as the one proposed in Hinton et al. [47] for training the deep belief networks, DATM is equipped with a SG-MCMC/VAE hybrid Bayesian inference algorithm that can jointly train all its hidden layers, as described in Algorithm 1. However, the same as most existing algorithms in deep learning, it still needs to specify the width of each layer,

In this paper, motivated by related work in Zhou et al. [18], [48], we adopt the idea of layer-wise training for DATM for the purpose of learning the width of each hidden layer in a greedy layer-wise manner, given a fixed budget on the width of the first layer. The proposed layer-wise training strategy is summarized in Algorithm 2. With a DATM of $L-1$ layers that has already been trained, the key idea is to use a truncated gamma-negative binomial process [4] to model the latent count matrix for the newly added top layer as $m_{kn}^{(L)(L+1)} \sim \text{NB}(r_k, p_n^{(L+1)})$, $r_k \sim \text{Gam}(\gamma_0/K_{Lmax}, 1/c_0)$, and rely on that stochastic process's shrinkage mechanism to prune inactive factors of layer L according to the values of $\{r_k\}_k$. Generally speaking, the inferred K_L would be clearly smaller than K_{Lmax} if K_{Lmax} is sufficiently large. As in Algorithm 2, K_{1max} is a parameter to set, whereas the inferred width of layer $l-1$, K_{l-1}, is set as the maximum number of factors of a newly added layer K_{lmax}. More details on this greedy layer-wise learning strategy can be found in Zhou et al. [18].

E. Variations of WHAI

To clearly understand how each component contributes to the overall performance of WHAI, below we consider some different variations.

Gamma hybrid autoencoding inference (GHAI): In the inference network of DATM, the reparameterizable Weilbull distribution is chosen to be the variational posterior and used to connect adjacent stochastic layers for the reasons specified in Section II-B. One may also choose some other distributions to construct the variational posterior. For example, one may replace (22) with

$$q(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{h}_n^{(l)}, \boldsymbol{\theta}_n^{(l+1)}) = \text{Gamma}(\boldsymbol{k}_n^{(l)} + \boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}, \boldsymbol{\lambda}_n^{(l)}). \quad (26)$$

While the gamma distribution does not have a simple reparameteriation, one may use RSVI [35] to define an approximate reparameterization procedure via rejection sampling. More specifically, following Naesseth et al. [35], to generate a gamma random variable $z \sim \text{Gamma}(\alpha, \beta)$, one may first use the rejection sampler [49] to generate $\tilde{z} \sim \text{Gamma}(\alpha + B, 1)$, for which the proposal distribution is expressed as

$$\tilde{z} = \left(\alpha + B - \frac{1}{3}\right)\left(1 + \frac{\varepsilon}{\sqrt{9(\alpha+B)-3}}\right)^3, \ \ \varepsilon \sim \mathcal{N}(0,1),$$

where B is a pre-set integer to make the acceptance probability be close to 1; one then lets $z = 1/\beta * \tilde{z}\prod_{i=1}^{B} u_i^{1/(\alpha+i-1)}$, where $u_i \sim \text{Uniform}(0,1)$. The gradients with respect to the ELBO, however, could still suffer from relatively high variance, as how likely a proposed ε will be accepted depends on the gamma distribution parameters, and B extra uniform random numbers $\{u_i\}_{1,B}$ need to be introduced.

Weibull autoencoding inference (WAI): To illustrate the effectiveness of the proposed hybrid Bayesian inference, we also consider WAI that has the same inference network as WHAI but infers $\{\boldsymbol{\Phi}^{(l)}\}_{1,L}$ and $\boldsymbol{\Omega}$ only using SGD. Although as argued in Mandt et al. [50], SGD can also be used for approximate Bayesian inference, and it performs well in AVITM [32], we will show in experiments that sampling the global parameters via TLASGR-MCMC provides improved performance in comparison to updating them via SGD.

Independent WHAI (IWHAI): To understand the importance of the stochastic-downward structure used in the inference network, we also consider IWHAI that remove the stochastic-downward connections of DATM-WHAI. More specifically, IWHAI redefines $q(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{h}_n^{(l)}, \boldsymbol{\theta}_n^{(l+1)})$ in (22) as Weibull$(\boldsymbol{k}_n^{(l)}, \boldsymbol{\lambda}_n^{(l)})$, and uses the same hybrid Bayesian inference to infer $\{\boldsymbol{\Phi}^{(l)}\}_{1,L}$ and $\boldsymbol{\Omega}$.

III. Supervised DATM for classification

With DATM, we are able to efficiently infer the topics of DLDA [17], [22] and directly project a document into its latent representation at multiple stochastic hidden layers, providing a new opportunity to learn interpretable latent representation that can well generate not only the observed bag of words of documents, but also the class labels that are often associated with documents. Thus, rather than following a two-step procedure to first apply DATM and then build a classifier on its unsupervisedly extracted latent features, we generalize DATM to a generative model for both the observed bags of words and labels, referred to as supervised DATM (sDATM), exploiting the synergy between document generation and classification to achieve enhanced performance.

Algorithm 2 Hybrid stochastic-gradient MCMC and autoencoding variational inference for DATM, which uses a layer-wise training strategy to train a set of networks, each of which adds an additional hidden layer on top of the previously inferred network, retrains all its layers jointly, and prunes inactive factors from the last layer.

Input: Observed data $\{\boldsymbol{x}_n\}_n$, upper bound of the width of the first layer K_{1max}, the number of layers L, the pruned threshold u, and hyper-parameters.
Output: Global parameters of DATM $\{\boldsymbol{\Phi}^{(l)}\}_{1,L}$ and $\boldsymbol{\Omega}$.
Set mini-batch size m;
for $l = 1, 2, \cdots L$ **do**
  Set K_{l-1}, the inferred width of layer $l-1$, as K_{lmax}, the upper bound of layer L's width.
  Initialize encoder parameters $\boldsymbol{\Omega}^{(l)}$ and decoder parameters $\boldsymbol{\Phi}^{(l)}$, combined with $\{\boldsymbol{\Omega}^{(t)}\}_{t=1}^{l-1}$ and $\{\boldsymbol{\Phi}^{(t)}\}_{t=1}^{l-1}$.
  for $iter = 1, 2, \cdots$ **do**
    Randomly select a mini-batch of m documents to form a subset $\mathbf{X} = \{\boldsymbol{x}_i\}_{1,m}$;
    Draw random noise $\{\varepsilon_i^t\}_{i=1,t=1}^{m,l}$ from uniform distribution;
    Calculate $\nabla_{\boldsymbol{\Omega}} L\left(\boldsymbol{\Omega}, \boldsymbol{\Phi}^{\{t\}}; \mathbf{X}, \varepsilon_i^t\right)$ according to (15), and update $\boldsymbol{\Omega}$;
    Sample $\left\{\boldsymbol{\theta}_i^{(t)}\right\}_{i=1,t=1}^{m,l}$ from (22) via $\boldsymbol{\Omega}$;
    for $l = 1, \cdots, L+1$ and $k = 1, \cdots, K_l$ **do**
      Update $M_k^{(l)}$ with (13), topics $\{\boldsymbol{\Phi}^{(l)}\}_{l=1}^{L}$ with (11), and $\boldsymbol{r}$ with (12).
    end for
  end for
  Delete the inactive topics of $\boldsymbol{\Phi}_k^{(l)}$ if $r_k < u$, and delete the corresponding paramters in $\boldsymbol{\Omega}^{(l)}$. Output the inferred width K_l.
end for

A. Label generation

We consider a labeled document corpus $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, where $y_n \in \{1, 2, \cdots, C\}$ and C is the total number of classes. We assume the label is generated from a categorical distribution $y_n \sim \text{Categorical}(p_{n1}, \ldots, p_{nC})$, where p_{nc} is the probability that $\boldsymbol{x}_n$ belongs to class c, which means

$$p(y_n) = \prod_{c=1}^{C} p_{nc}^{\delta(y_n=c)}, \tag{27}$$

where $\delta(\cdot)$ is an indicator function that is equal to one if the argument is true and zero otherwise.

In a usual supervised-learning setting that maps an observation to its label via a deterministic deep NN, it is often only the features at the top hidden layer (furthest from the data) that are transformed to define the label probabilities p_{nc}. For DATM, as the latent representation $\boldsymbol{\theta}_n^{(l)}$ at different hidden layers are stochastically connected, the topics at different stochastic layers reveal different levels of abstraction, and it is the features at the bottom hidden layer (closest to the data) that are directly responsible for data generation, we are motivated to concatenate $\boldsymbol{\theta}_n^{(l)}$ across all hidden layers to construct a latent feature vector as

$$\boldsymbol{s}_n = \left[\boldsymbol{\theta}_n^{(1)}, \cdots, \boldsymbol{\theta}_n^{(L)}\right]. \tag{28}$$

With this concatenation, the label information is directly used to influence the features across all layers, which helps improve the discrimination power and robustness of the learned features [51].

To map from $\boldsymbol{s}_n$ to its label probability vector $\boldsymbol{p}_n = (p_{n1}, \ldots, p_{nC})$, we first consider a linear setting that lets

$$\boldsymbol{p}_n = \left[\frac{e^{\boldsymbol{w}_1^T \boldsymbol{s}_n}}{\sum_{c=1}^C e^{\boldsymbol{w}_c^T \boldsymbol{s}_n}}, \cdots, \frac{e^{\boldsymbol{w}_C^T \boldsymbol{s}_n}}{\sum_{c=1}^C e^{\boldsymbol{w}_c^T \boldsymbol{s}_n}}\right], \tag{29}$$

where $\mathbf{W}_c = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_C]$ can be considered as the coefficients of a linear classifier, whose features are the concatenation of the latent features projected from $\boldsymbol{x}_n$ using (22).

Note although the features $\boldsymbol{s}_n$ in (29) is nonlinear transformed from $\boldsymbol{x}_n$, those nonlinear mappings are primarily used to approximate the posterior of $\{\boldsymbol{\theta}_n^{(l)}\}$. To further boost the performance of classification, L layer-specific multi-layer perceptrons (MLPs) $\{g_1^{(l)}\}_{l=1}^L$ are used to map layer-specific feature spaces to a concatenated feature space as

$$\boldsymbol{s}_n = \left[g_1^{(l)}(\boldsymbol{\theta}_n^{(1)}), \cdots, g_1^{(L)}(\boldsymbol{\theta}_n^{(L)})\right]. \tag{30}$$

Then, another MLP g_2 is used to transform the concatenated features $\boldsymbol{s}_n$ to the probabilistic space as

$$\boldsymbol{p}_n = \left[\frac{e^{\boldsymbol{w}_1^T g_2(\boldsymbol{s}_n)}}{\sum_{c=1}^C e^{\boldsymbol{w}_c^T g_2(\boldsymbol{s}_n)}}, \cdots, \frac{e^{\boldsymbol{w}_C^T g_2(\boldsymbol{s}_n)}}{\sum_{c=1}^C e^{\boldsymbol{w}_c^T g_2(\boldsymbol{s}_n)}}\right]. \tag{31}$$

We use $\mathbf{W}_m$ to denote all parameters in MLPs. No matter for the linear model or the nonlinear one, the label likelihood (27) can be rewritten as $p(y_n \mid \{\boldsymbol{\theta}_n^{(l)}\}_{l=1}^L)$, resulting in a fully-generative model for $\{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$ as shown in Fig. 1d. The linear model and the nonlinear one are represented as sDATM-L and sDATM-N, respectively, whose inference models are the same with DATM shown in Fig. 1a.

B. Model learning and prediction

With the generative process of sDATM, we can write the ELBO of $p(\boldsymbol{x}, \boldsymbol{y})$ as

$$\begin{aligned} L = &\sum_{n=1}^{N} \mathbb{E}\left[\ln p\left(\boldsymbol{x}_n \mid \boldsymbol{\Phi}^{(1)}, \boldsymbol{\theta}_n^{(1)}\right) + \ln p\left(\boldsymbol{y}_n \mid \{\boldsymbol{\theta}_n^{(l)}\}_{l=1}^L\right)\right] \\ &- \sum_{n=1}^{N}\sum_{l=1}^{L} \mathbb{E}\left[\ln \frac{q\left(\boldsymbol{\theta}_n^{(l)}\right)}{p\left(\boldsymbol{\theta}_n^{(l)} \mid \boldsymbol{\Phi}^{(l+1)}, \boldsymbol{\theta}_n^{(l+1)}\right)}\right] \\ &- \sum_{c=1}^{C} KL\left[q(\boldsymbol{w}_c) || p(\boldsymbol{w}_c)\right], \end{aligned} \tag{32}$$

where the expectations are taken with respect to $q\left(\{\boldsymbol{\theta}_n^{(l)}\}_{n=1,l=1}^{N,L}\right)$, modeled by (22), and $q(\{\boldsymbol{w}_c\}_{c=1}^C)$. The prior and the variational posterior of $\{\boldsymbol{w}_c\}_{c=1}^C$ are set as diagonal Gaussian distributions [52] [53] as

$$\begin{aligned} p(\boldsymbol{w}_c) &= \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ q(\boldsymbol{w}_c) &= \mathcal{N}(\boldsymbol{\mu}_c, \text{diag}(\boldsymbol{\sigma}_c)), \end{aligned} \tag{33}$$

resulting in an analytic KL divergence as

$$\mathrm{KL}\left[q(\boldsymbol{w}_c)||p(\boldsymbol{w}_c)\right] = \frac{1}{2}\left(||\boldsymbol{\mu}_c||_2^2 + ||\boldsymbol{\sigma}_c||_2^2\right) - \log||\boldsymbol{\sigma}_c||, \quad (34)$$

which can be viewed as a prior regularization on $\boldsymbol{w}_c$. To ensure $\boldsymbol{\sigma}_c$ to be nonnegative, we parameterize it pointwise as $\boldsymbol{\sigma}_c = \log(1+\exp(\boldsymbol{\rho}_c))$ and update the variational parameters $\{\boldsymbol{\mu}_c, \boldsymbol{\rho}_c\}_{c=1}^C$ with a usual backpropagation algorithm. For nonlinear sDATM, the network structure of $\mathbf{W}_m$ in (30) and (31) is set to:

$$\begin{aligned} g_1^{(l)}(\boldsymbol{\theta}_n^{(l)}) &= \ln[1+\exp(\mathbf{W}_{m1}^{(l)}\boldsymbol{\theta}_n^{(l)} + \boldsymbol{b}_{m1}^{(l)})], \\ \boldsymbol{h}_n &= \ln[1+\exp(\mathbf{W}_{m2}\boldsymbol{s}_n + \boldsymbol{b}_{m2})], \\ g_2(\boldsymbol{s}_n) &= \ln[1+\exp(\mathbf{W}_{m3}\boldsymbol{h}_n + \boldsymbol{b}_{m3})], \end{aligned} \quad (35)$$

where $\{\mathbf{W}_{m1}^{(l)}\}_{l=1}^L \in \mathbb{R}^{K_l \times K_l}$, $\{\mathbf{W}_{m2}\} \in \mathbb{R}^{a_1 \times \sum_l K_l}$, $\{\mathbf{W}_{m3}\} \in \mathbb{R}^{a_2 \times a_1}$, $\{\boldsymbol{b}_{m1}^{(l)}\}_{l=1}^L \in \mathbb{R}^{K_l}$, $\boldsymbol{b}_{m2} \in \mathbb{R}^{a_1}$, and $\boldsymbol{b}_{m3} \in \mathbb{R}^{a_2}$, with $a_1 = 400$, $a_2 = 200$ in the experiments.

With the inferred variational parameters of the inference network, at the test stage, approximating the intractable expectation $\mathbb{E}_{q(\boldsymbol{s},\boldsymbol{w}_{1:C}\,|\,\boldsymbol{x})}[p(y\,|\,\boldsymbol{s},\boldsymbol{w}_{1:C})]$ with Monte Carlo estimation, we can predict the label of a testing document as

$$y = \mathrm{argmax}_c\left(\textstyle\sum_{j=1}^{N_{collect}} p(y=c\,|\,\boldsymbol{w}_1^{(j)}, \boldsymbol{s}^{(j)})\right)_{c=1,C}, \quad (36)$$

where $\boldsymbol{w}_c^{(j)} \sim q(\boldsymbol{w}_c)$ and $\boldsymbol{\theta}^{(j)} \sim q(\boldsymbol{\theta})$ accord to (33) and (22), respectively, $\boldsymbol{s}^{(j)}$ is deterministically transformed from $\boldsymbol{\theta}^{(j)}$, and $N_{collect} = 50$ is used.

IV. Experimental results

In this paper, DATM is proposed for extracting deep latent features and analyzing documents unsupervisedly, and sDATM is proposed for joint deep topic modeling and document classification. In this section, the performance of the proposed models are demonstrated through both unsupervised and supervised learning tasks on big corpora. Our code is written based on Theano [54].

A. Unsupervised learning for document representation

1) Per-heldout-word perplexity: We first compare the per-heldout-word perplexity [13], [17], [55], a widely-used performance measure, of different models on 20Newsgroups (20News), Reuters Corpus Volume I (RCV1), and Wikipedia (Wiki). 20News consists of 18,845 documents with a vocabulary size of 2,000. RCV1 consists of 804,414 documents with a vocabulary size of 10,000. Wiki, with a vocabulary size of 7,702, consists of 10 million documents randomly downloaded from Wikipedia using the script provided by Hoffman et al. [56]. For Wiki, we randomly select 100,000 documents for testing, and to be consistent with previous settings [13], [22], [55], no precautions are taken in the Wikipedia downloading script to prevent a testing document from being downloaded into a mini-batch for training.

For comparison, the models included in our comparison are listed as follows, using the code provided by the authors:

- **LDA**: Latent Dirichlet allocation [1] is a basic probability topic model, which is closely related to a single-hidden-layer version of DLDA. We run it by onlineVB.
- **OR-softmax**: Over-replicated softmax [10] is a type of deep Boltzmann machine that is suitable for extracting distributed semantic representation from documents.
- **DocNADE**: Document neural autoregressive distribution estimation [57] is an autoregressive distribution estimator based on feed-forward NNs for text analysis.
- **DPFA**: Deep Poisson factor analysis [13] is a hierarchical model for text analysis based on Poisson factor analysis and sigmoid belief network.
- **AVITM**: Autoencoding variational inference for topic modeling [32] is an autoencoding topic model based on a single-hidden-layer LDA.
- **DLDA-Gibbs** and **DLDA-TLASGR**: Deep latent Dirichlet allocation inferred by Gibbs sampling [18] and by TLASGR-MCMC [22], respectively.

In order to further demonstrate the advantages of the stochastic upward-downward structure and hybrid inference algorithm, some variants including **DATM-GHAI**, **DATM-WAI** and **DATM-IWHAI** discussed in Section II-E are also included for comparison. Note that as shown in Cong et al. [22], DLDA-Gibbs and DLDA-TLASGR are state-of-the-art topic modeling algorithms that outperform a large number of previously proposed ones, such as deep Poisson factor modeling [55] and the nested hierarchical Dirichlet process [58].

Similar to previous work [14], [59], [60], for each corpus, we randomly select 70% of the word tokens from each document to form a training matrix $\mathbf{T}$, holding out the remaining 30% to form a testing matrix $\mathbf{Y}$. We use $\mathbf{T}$ to train the model and calculate the per-heldout-word perplexity as

$$\exp\left\{-\frac{1}{y_{\cdot\cdot}}\sum_{v=1}^{V}\sum_{n=1}^{N} y_{vn}\ln\frac{\sum_{s=1}^{S}\sum_{k=1}^{K^1}\phi_{vk}^{(1)s}\theta_{kn}^{(1)s}}{\sum_{s=1}^{S}\sum_{v=1}^{V}\sum_{k=1}^{K^1}\phi_{vk}^{(1)s}\theta_{kn}^{(1)s}}\right\}, \quad (37)$$

where S is the total number of collected samples and $y_{\cdot\cdot} = \sum_{v=1}^{V}\sum_{n=1}^{N} y_{vn}$. For the proposed models, we set the mini-batch size as 200, and use 2000 mini-batches for burn-in on both 20News and RCV1 and 3500 on Wiki. We collect 3000 samples after burn-in to calculate perplexity. The hyperparameters of WHAI are set as: $\eta^{(l)} = 1/K_l$, $\boldsymbol{r} = \mathbf{1}$, and $c_n^{(l)} = 1$.

Table I lists for various algorithms both the perplexity and the average run time per testing document given 3000 random samples of the global parameters. For fair comparison, all the models are evaluated on the same 3.0 GHz CPU. We first compare different DLDA based models and then compare the proposed DATM-WHAI with other non-DLDA based models.

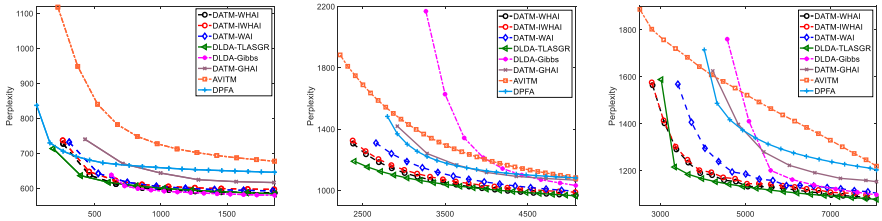
Given the same generative network structure, DLDA-Gibbs performs the best in terms of predicting heldout word tokens, which is not surprising as this batch algorithm can sample from the true posteriors given a sufficiently large number of Gibbs sampling iterations. DLDA-TLASGR is a mini-batch algorithm that is much more scalable in training than DLDA-Gibbs, at the expense of slightly degraded performance in out-of-sample prediction. Both DATM-WAI, using SGD to infer the global parameters, and DATM-WHAI, using a stochastic-gradient MCMC to infer the global parameters, slightly underperform DLDA-TLASGR. Compared with DATM-GHAI approximately reparameterizing the gamma distributions, DATM-WHAI that has simple reparameterizations for its Weibull distributions out-

TABLE I: Comparisons of per-heldout-word perplexity and testing time (average seconds per document, with 3000 random samples) on three different datasets.

Model	Size	Perplexity			Test Time		
		20News	RCV1	Wiki	20News	RCV1	Wiki
LDA	128	593	1039	1059	4.14	11.35	12.16
OR-softmax	128-64-32	592	1013	1024	3.20	8.64	9.77
DocNADE	128	591	969	999	0.42	0.90	1.04
DPFA	128-64-32	637	1041	1056	20.12	34.21	35.41
AVITM	128	654	1062	1088	0.23	0.68	0.80
DLDA-Gibbs	128-64-32	571	938	966	10.46	23.38	23.69
DLDA-Gibbs	128-64	573	942	968	8.73	18.50	19.79
DLDA-Gibbs	128	584	951	981	4.69	12.57	13.31
DLDA-TLASGR	128-64-32	579	950	978	10.46	23.38	23.69
DLDA-TLASGR	128-64	581	955	979	8.73	18.50	19.79
DLDA-TLASGR	128	590	963	993	4.69	12.57	13.31
DATM-GHAI	128-64-32	604	963	994	0.66	1.25	1.49
DATM-GHAI	128-64	608	965	997	0.44	0.96	1.05
DATM-GHAI	128	615	972	1003	0.22	0.69	0.80
DATM-IWHAI	128-64-32	588	964	990	0.58	1.15	1.38
DATM-IWHAI	128-64	589	965	992	0.38	0.87	0.97
DATM-IWHAI	128	592	966	996	0.20	0.66	0.78
DATM-WAI	128-64-32	581	954	984	0.63	1.20	1.43
DATM-WAI	128-64	583	958	986	0.42	0.91	1.02
DATM-WAI	128	593	967	999	0.20	0.66	0.78
DATM-WHAI	128-64-32	581	953	980	0.63	1.20	1.43
DATM-WHAI	128-64	582	957	982	0.42	0.91	1.02
DATM-WHAI	128	591	965	996	0.20	0.66	0.78

TABLE II: Comparisons of per-heldout-word perplexity by layer-wise training strategy to infer the network structure (the same settings with Table I) on three different datasets.

K_{1max}	Inferred structure			Perplexity		
	20News	RCV1	Wiki	20News	RCV1	Wiki
64	64-62-55	64-64-61	64-64-59	584	959	987
128	121-110-84	126-118-102	123-114-96	578	949	978
256	248-211-183	253-220-196	250-217-188	574	943	972
512	470-197-155	482-201-167	471-193-170	574	941	971
1024	478-199-160	484-201-163	472-190-174	573	940	971



(a) (b) (c)

Fig. 3: Plot of per-heldout-word perplexity as a function of time for (a) 20News, (b) RCV1, and (c) Wiki. Except for AVITM that has a single hidden layer with 128 topics, all the other algorithms have the same network size of 128-64-32 for their deep generative models.

performs DATM-GHAI. Besides, thanks to the use TLASGR-MCMC rather than a simple SGD procedure, DATM-WHAI consistently outperforms DATM-WAI. It is also clear that except for DATM-IWHAI that has no stochastic-downward components in its inference, all the other variations of DATM have a clear trend of improvement as the generative network becomes deeper, indicating the importance of having stochastic downward information propagation during posterior inference. Compared with DLDA-Gibbs and DLDA-TLASGR that need to perform Gibbs sampling at the testing stage, DATM-WHAI and its variations are considerably faster in processing a testing document, due to the use of an inference network.

Further, comparing DATM-WHAI with the methods in the first group in Table I shows that all algorithms with an inference network, including AVITM, DocNADE, and DATM-WHAI, clearly outperform those relying on an iterative procedure for out-of-sample prediction, including OR-softmax, LDA, and DPFA. In terms of perplexity, it can be seen that DATM-WHAI with a single hidden layer already clearly outperforms AVITM, indicating that using the Weibull distribution is more appropriate than using the logistic normal distribution to model the document latent representation. Compared with DocNADE, an outstanding autoregressive and shallow model, the single-layer DATM-WHAI with the same number of topics

marginally improves the perplexity and test speed. Distinct from DocNADE, DATM-WHAI is able to add more stochastic hidden layers to extract hierarchical topic representations and further improve its perplexity, and its non auto-regressive structure makes it easier to be accelerated with GPUs.

As discussed in Section II-D, DATM is able to infer the network structure via a greedy layer-wise training strategy given a fixed budget on the width of the first layer. We perform experiments with $L = 3$, $K_{1max} \in \{64, 128, 256, 512, 1024\}$, and the pruning threshold as $u = 0.01$. Shown in Table II are the inferred network structure and perplexities over three different corpora. We observe a clear trend of improvement by increasing K_{1max} until saturation (when K_{1max} becomes sufficiently large). Moreover, when $K_{1max} = 128$, in comparison to the results of a fixed 128-64-32 network structure shown in Table I, we find that a better network structure with lower perplexity is inferred, illustrating the effectiveness of our proposed method.

Below we examine how various inference algorithms progress over time during training, evaluated with per-holdout-word perplexity. As clearly shown in Fig. 3, DATM-WHAI outperforms DPFA and AVITM in providing lower perplexity as time progresses, which is not surprising as the DLDA multi-layer generative model is good at document representation, while AVITM is only "deep" in the deterministic part of its inference network and DPFA is restricted to model binary topic usage patterns via its deep network. When DLDA is used as the generative model, in comparison to Gibbs sampling and TLASGR-MCMC on two large corpora, RCV1 and Wiki, the mini-batch based WHAI converges slightly slower than TLASGR-MCMC but much faster than Gibbs sampling; WHAI consistently outperforms WAI, which demonstrates the advantage of our proposed hybrid Bayesian inference algorithm; in addition, the RSVI based DATM-GHAI clearly converges more slowly in time than DATM-WHAI does. Note that for all three datasets, the perplexity of TLASGR decreases at a fast rate, followed closely by that of WHAI, while that of Gibbs sampling decreases slowly, especially for RCV1 and Wiki, as shown in Figs. 3b and 3c. This is expected as both RCV1 and Wiki are much larger corpora, for which a mini-batch based inference algorithm can already make significant progress in inferring the global model parameters, before a batch-learning Gibbs sampler finishes a single iteration that needs to go through all documents. We also notice that although AVITM is fast for testing via the use of a VAE, its representation power is limited due to not only the use of a shallow topic model, but also the use of a latent Gaussian based inference network that is not naturally suited to model document latent representation.

2) Topic hierarchy: In addition to quantitative evaluations, we have also visually inspected the inferred topics at different layers and the inferred connection weights between the topics of adjacent layers. Distinct from many existing deep topic models that build nonlinearity via "black-box" NNs, we can easily visualize the whole stochastic network, whose hidden units of layer $l-1$ and those of layer l are connected by $\phi^{(l)}_{k'k}$ that are sparse. In particular, we can understand the meaning of each hidden unit by projecting it back to the original data space via $\left[\prod_{t=1}^{l-1} \mathbf{\Phi}^{(t)}\right] \boldsymbol{\phi}^{(l)}_k$, as described in Section II-A. We show in Fig. 5 a subnetwork, originating from Topics (units) 16, 19, and 24 of the top hidden layer, taken from the generative network of size 128-64-32 inferred on Wiki. Note plotting the whole network at once is often unrealistic and hence we resort to extracting a subnetwork for visualization. The reason that these three topics are combined as the roots to form the subnetwork shown in Fig. 5 is because they share similar key words and appear somewhat related to each other. Both the semantic meanings of the inferred topics and the connection weights between them are highly interpretable. These topics tend to be very specific at the bottom layer, and become increasingly more general at higher layers. Note that the higher-layer topics gather the general semantics from their leaf nodes, leading to the fact that some words may appear with large weights in several different higher-layer topics. For example, in Fig. 5, both Topics 16 and 19 at layer 3 talk about "international/group/company," but Topic 16 pays more attention to "business" while Topic 19 focuses more on "organization." Several additional example topic subnetworks rooted at different top-layer nodes are shown in Figs. 1, 2, 3, and 6 in the Supplement. Moreover, comparisons of the hierarchical structures learned by hLDA [15], DEF [16], and DATM on 20News and NIPS12[1] are provided in the Supplement, which clearly demonstrate the unique hierarchical topic structure and its interpretability under the proposed model. More discussions can be found in the Supplement.

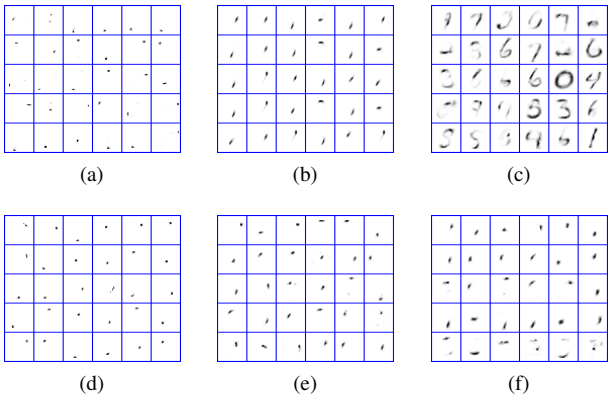


Fig. 4: Learned topics on MNIST digits with a three-hidden-layer DATM of size 128-64-32. Shown in (a)-(c) are example topics for layers 1, 2 and 3, respectively, learned with a deterministic-upward-stochastic-downward encoder (DATM-WHAI), and shown in (d)-(f) are the ones learned with a deterministic-upward encoder (DATM-IWHAI).

To further illustrate the effectiveness of the multi-layer representation in our model, we apply a three-layer DATM to MNIST digits and present the learned dictionary atoms. We use the Poisson likelihood directly to model the MNIST digit pixel values that are nonnegative integers ranging from 0 to 255. As shown in Figs. 4a-4c, it is clear that the factors at layers one to three represent localized points, strokes, and digit components,

[1] http://www.cs.nyu.edu/~roweis/data.html

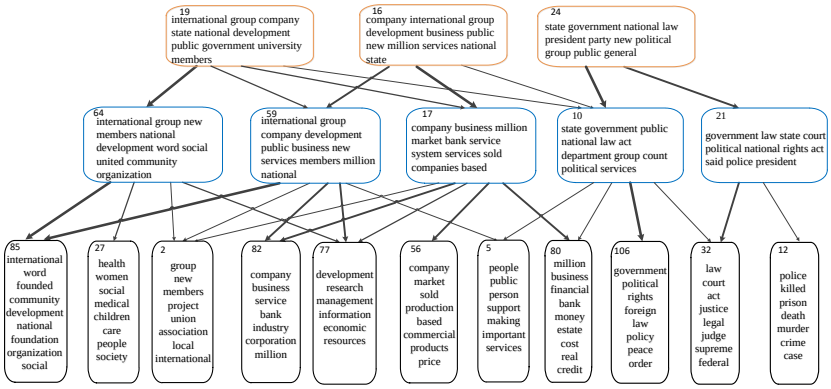


Fig. 5: Example of hierarchical topics learned from Wiki by a three-hidden-layer DATM-WHAI of size 128-64-32.

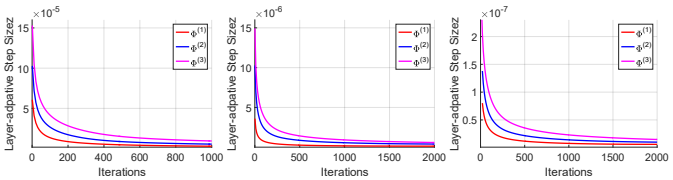


Fig. 6: Topic-layer-adaptive learning rates inferred with a three-layer DATM of size 128-64-32. (a) 20News. (b) RCV1. (c) Wiki

Fig. 7: Latent space interpolations on the MNIST test set. Left and right columns correspond to the images generated from $\boldsymbol{z}_1^{(3)}$ and $\boldsymbol{z}_2^{(3)}$, and the others are generated from the latent representations interpolated linearly from $\boldsymbol{z}_1^{(3)}$ to $\boldsymbol{z}_2^{(3)}$.

respectively, that cover increasingly larger spatial regions. This type of hierarchical visual representation is difficult to achieve with other types of deep NNs [10], [27]–[29]. WUDVE, the inference network of WHAI, has a deterministic-upward–stochastic-downward structure, in contrast to a conventional VAE that often has a pure deterministic bottom-up structure. Here, we further visualize the importance of the stochastic-downward part of WUDVE through a simple experiment. We remove the stochastic-downward part of WUDVE in (22) represented as DATM-IWHAI, in other words, we ignore the top-down information. As shown in Figs. 4d-4f, although some latent structures are learned, the hierarchical relationships between adjacent layers almost all disappear, indicating the importance of having a stochastic-downward structure together with a deterministic-upward one in the inference network.

3) Topic-layer-adapative stepsize: To illustrate the working mechanism of our proposed topic-layer-adapative stepsize, we show how its inferred learning rates are adapted to different layers in Fig. 6, which is obtained by averaging over the learning rates of all $\phi_k^{(l)}$ for $k = 1, \cdots, K_l$. For $\boldsymbol{\Phi}^{(l)}$, higher layers prefer larger step sizes, which may be attributed to the enlarge-partition-augment data generating mechanism of DLDA. In addition, we find that larger datasets prefer slower learning rates, which demonstrates that since the stochastic noise brought by minibatch learning increases, the model needs a smaller learning rate.

4) Topic manifold: As a sanity check for whether DATM overfits the data, we show in Fig. 7 the latent space interpolations between the test set examples on MNIST dataset, and provide related results in the Supplement for the 20News corpus. In Fig. 7, the leftmost column is from the same image represented as $\boldsymbol{x}_1$ and the rightmost column is random sampled from a class represented as $\boldsymbol{x}_2$. With the 3-layer model learned before, following Dumoulin et al. [61], $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are projected into $\boldsymbol{z}_1^{(3)}$ and $\boldsymbol{z}_2^{(3)}$. We then linearly interpolate between $\boldsymbol{z}_1^{(3)}$ and $\boldsymbol{z}_2^{(3)}$, and pass the intermediate points through the generative model to generate the input-space interpolations, shown in the middle columns. We observe smooth transitions between the examples in all pairs, and the intermediate images remain interpretable. These observations suggest that the inferred latent space of the model resides on a manifold and WHAI has learned a generalizable latent feature representation rather than concentrating its probability mass around the training examples.

B. Supervised feature learning for classification

To evaluate how well sDATM leverages the label information for feature learning, we compare its classification performance with a variety of algorithms on both MNIST digits and several benchmark datasets for text classification. For all experiments, the first 100 epochs are used to train DATM without the label information, and then another 300 epochs are used to train sDPATM with the label information. Note as the ELBO in (32) contains several KL regularization terms, following Sonderby et al. [29], warm-up is used during the first several epochs to gradually impose the KL regularization terms.

Digit classification. We first test sDATM on the MNIST dataset, which consists of 60,000 training handwritten digits and 10,000 testing ones. We list the results in Table III. Among them, DLDA inferred by Gibbs sampling and Gaussin VAE inferred by SGD are the unsupervised feature learning

TABLE III: Error rates (%) and testing time (average seconds per image) on MNIST dataset.

Model	Error Rate	Test Time
DLDA+SVM [17]	2.82	0.523
VAE+SVM [62]	1.04	0.081
DBN [47]	1.20	0.021
FNN [64]	1.14	0.013
MMVA [62]	0.90	0.014
AAE [63]	0.85	0.015
sDATM-L	1.03	0.011
sDATM-N	0.97	0.013

models, which are followed by a linear SVM, represented as DLDA+SVM and VAE+SVM, respectively. Other models, including a supervised VAE model under Gaussian assumption called max-margin variational autoencoder (MMVA) [62], a supervised generative adversarial network called Adversarial Autoencoders (AAE) [63], a fully-connected NN (FNN) [64], and the deep belief network (DBN) [47], are also compared. In addition, except for sDATM and DLDA that use the original gray-scale pixel values from 0 to 255, all the other algorithms divide them by 255, normalizing them to nonnegative real values from 0 to 1.

Shown in Table III are the error rates of various algorithms, which are provided in their corresponding papers, except for that of DLDA+SVM which is obtained by running the author provided code; the test times are obtained by running the author provided code on the same computer with a 3.0 GHz CPU. Since both sDATM-L and sDATM-N extract features and learn classifier jointly by a principled fully-generative model, it is unsurprising that their test errors are clearly lower than that of DLDA+SVM. Meanwhile, the nonlinear sDATM-N only slightly outperforms the linear sDATM-L, demonstrating the effectiveness of sDATM in transforming the data into a discriminative latent space. In addition, thanks to the encoder network in sDATM, it takes substantial less time than DLDA+SVM does at the testing stage. In contrast to both DBN and FNN that use deterministic "black-box" deep NNs, sDATM learns an interpretable multi-stochastic-layer latent space and provides a lower testing error. MMVA and AAE perform slightly better but takes longer time at the testing stage than sDATM does, which may be attributed to more complex networks used in them.

Shown in Fig. 8a are how the test errors of sDATM change as the layer width and network depth vary. There is a clear trend of test error reduction as the network depth of sDATM increases, suggesting the effectiveness of having a multi-layer representation and feature fusion. When the network depth is fixed, sDATM with a larger network width performs better, suggesting sDATM is able to use a larger capacity network to learn a more discriminative latent space.

Document classification. We also test sDATM on the following three document classification tasks: 20News, RCV1, and IMDB [65]. Different from the perplexity experiments, a lager vocabulary of 33,420 words is used for 20News to achieve better performance [18]. Since RCV1 has 103 topic categories in a hierarchy and one document may be associated with more than one topic, we transform them to a single-

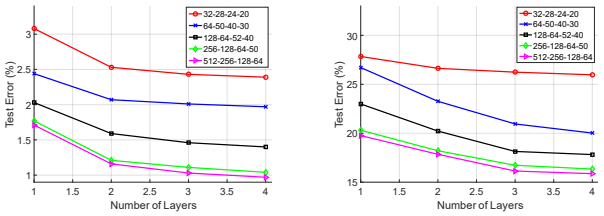


Fig. 8: The test errors change with different layers and width on (a) MNIST and (b) 20News by sDATM-N.

TABLE IV: Test error rates (%) and testing time (average seconds per document) on 20News, RCV1, and IMDB datasets.

Model	Error Rate			Test Time		
	20News	RCV1	IMDB	20News	RCV1	IMDB
LDA	25.40	24.17	21.46	0.60	0.25	0.49
DLDA	22.01	20.18	18.13	1.22	0.92	1.06
DocNADE	23.21	21.07	18.79	0.031	0.019	0.024
OR-softmax	22.05	20.19	18.24	0.69	0.21	0.58
AVITM	26.31	25.16	21.75	0.017	0.014	0.015
DATM	24.28	23.10	20.42	0.018	0.015	0.016
FNN-BOW	31.29	28.16	19.25	0.014	0.010	0.012
FNN-tfidf	24.16	19.28	17.14	0.013	0.009	0.010
sAVITM	20.15	18.61	16.13	0.015	0.012	0.014
MedLDA	18.76	16.38	15.28	0.240	0.098	0.202
wv-LSTM	18.00	16.04	13.50	-	-	-
sDATM-L	18.63	15.42	13.66	0.016	0.013	0.014
sDATM-N	15.81	13.40	10.92	0.018	0.014	0.015

label classification with 55 different classes as discussed in Johnson & Zhang [66], [67]. The IMDB dataset is used for sentiment classification on movie reviews with a vocabulary size of 30,000 after preprocessing. The task is to determine whether the movie reviews are positive or negative. For a fair comparison, the training/testing random splits follow the same settings in previous work [18], [66], [67].

Besides a number of unsupervised models, including LDA, DLDA, DocNADE, OR-softmax, and AVITM, we also make comparison to several representative supervised models:

- **FNN-BOW** and **FNN-tfidf**: Four-layer fully-connect feed-forward NNs (FNN) of size 512-256-128-64 with bag-of-words (BOW) and tfidf document features.
- **MedLDA**: Gibbs max-margin supervised topic models [68] based on LDA.
- **sAVITM**: A supervised AVITM through adding a linear softmax classifier to AVITM.
- **wv-LSTM**: A LSTM model based on word vector sequence [69].

For DLDA and sDATM, we choose a four-layer structure with the size of 512-256-128-64.

Listed in Table IV are the results for various algorithms, where these of wv-LSTM are provided in Dai & Le [69], these for FNN-bow and FNN-tfidf are obtained by our own carefully optimized code, and all the others are obtained by running the author provided code, all on the same data used by sDATM. Although DLDA achieves the lowest testing errors among all unsupervised models, which illustrates the effectiveness of the multi-layer representation of DLDA, it underperforms all supervised topic models. Having the same

musical music dance song acting	war soldier army battle fight	wife husband man woman life	airship rocket star launch spacecraft	society culture social modern cultural
funny fun charming well enjoyment	interesting interested like love enjoy	meaningful intriguing attractive attract expect	bad bored boring sad tired	worse worst vapid tiresome ridiculous

Fig. 9: The top five first-layer topics learned by DATM (the first row) and those by sDATM (the second row).

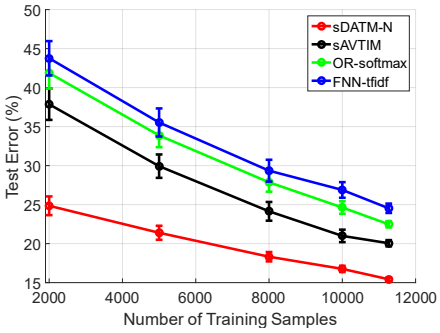


Fig. 10: The test errors change with different sizes of training dataset on 20News.

document features (BOW) and similar encoder structures, supervised FNN-BOW underperforms DLDA, indicating that the interpretable latent feature space of DLDA is amenable to classification. Both sDATM-L and sDATM-N, which integrate feature extraction and classification via a fully generative model for both documents and labels, clearly outperform DLDA. sDATM-L outperforms both sAVITM and MedLDA, which suggests that its multilayer latent representation provides more discriminative power. sDATM-N further improves over sDATM-L by introducing nonlinearity into the mapping from the latent features to labels. Note LSTM is a popular method to model the sequence information between words in a document. The proposed sDATM-L and sDATM-N, only using BOW features, achieve comparable or better performance in comparison to wv-LSTM. In Fig. 8b, we also show on the 20News dataset how the test errors of sDATM-N change as the network depth and width vary.

Supervised topic modeling. It can be noted from (32) that the topics and latent representations of sDATM are related to not only the documents but also their labels, which is why sDATM is able to provide more discriminative power than DATM. For illustration, we compare the top five first-layer topics learned by DATM and that by sDATM-L on the IMDB dataset in Fig. 9. The documents in IMDB contain both movie descriptions and viewer comments. Clearly, the top topics inferred by DTAM are focused on the content of the movies, while these by sDATM are focused on the sentiments of the viewers, which helps explain why sDATM performs much better than DATM in document classification for IMDB.

In order to better understand the changes of topics from DATM to sDATM, we first train DATM with 100 epochs and then add label information to train sDATM with another 100 epochs. Figs. 11 and 12 show how one topic tree changes by changing the model from DATM to sDATM. Clearly, some connection weights between the topics of adjacent layers change and some topics become more focused on viewer sentiments after adding the label likelihood.

Robust to the smaller training set. Compared with the deterministic mapping in DNNs, sDATM constructs a probabilistic model to perform distribution estimation, which may bring more robustness to the smaller training set. In order to demonstrate this advantage, we train the models on 20News with different training data sizes, as shown in Fig. 10 achieved by 50 independent experiments. Due to the relative small training dataset, deterministic FNN with tfidf document features performs worst among all the models. The parameters in the generative model of sAVITM (the topics) and OR-softmax (the weights) are updated by SGD with a point estimate, which results in an obvious increase in test error. In addition, the variance of sDATM is smaller than that of others especially when the training data is small. We attribute it to the following three reasons: 1) the fully distribution estimate of sDATM, no matter for the networks' parameters or the classifiers; 2) the model average when it perform prediction; and 3) the robustness brought by the multi-layer feature fusion.

V. Conclusion

We propose an interpretable deep generative model for document analysis, referred to as deep autoencoding topic model (DATM), where deep latent Dirichlet allocation is used as the generative network, and a Weibull upward-downward variational encoder is used to approximate the posterior distribution of the latent representation. Scalable Bayesian inference for DATM is realized by a hybrid stochastic gradient MCMC and variational inference algorithm. We further construct supervised DATM that can jointly model the documents and their labels. The efficacy and scalability of the proposed models are demonstrated on a variety of unsupervised and supervised learning tasks with big corpora.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
[2] J. D. Lafferty and D. M. Blei, "Correlated topic models," in *Advances in Neural Information Processing Systems*, 2005, pp. 147–154.
[3] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *J. Amer. Statist. Assoc.*, 2006.
[4] M. Zhou and L. Carin, "Negative binomial process count and mixture modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 307–320, 2015.
[5] J. Zhu, A. Ahmed, and E. P. Xing, "MedLDA: Maximum margin supervised topic models," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2237–2278, 2012.
[6] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," in *Advances in Neural Information Processing Systems*, 2012, pp. 2222–2230.
[7] C. Wang, B. Chen, and M. Zhou, "Multimodal Poisson gamma belief network," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
[8] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Knowledge Discovery and Data Mining, 2011.*

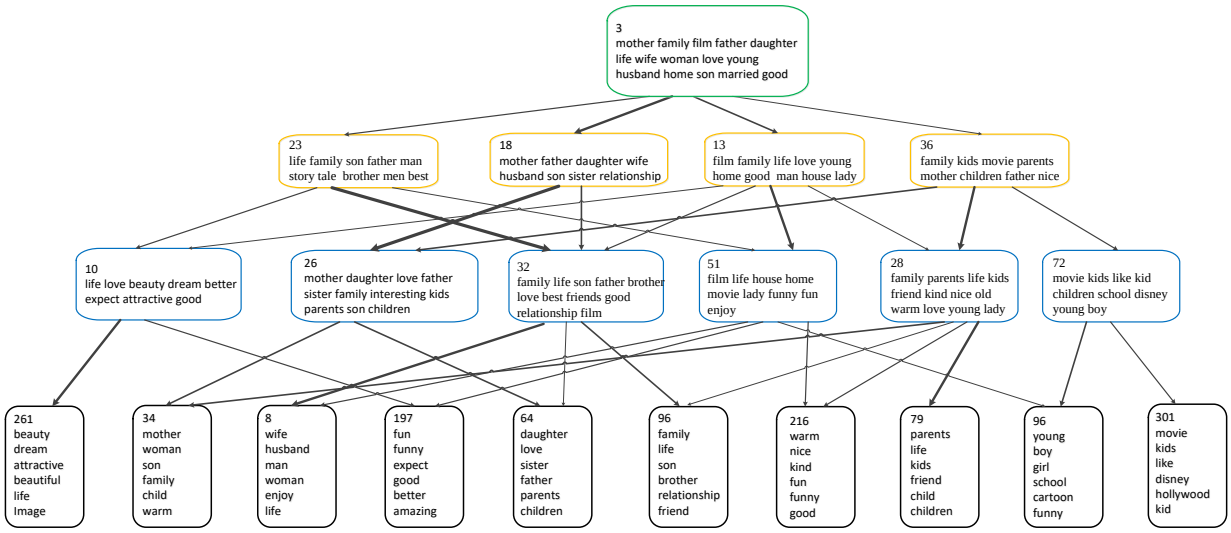


Fig. 11: Topics by unsupervised learning.

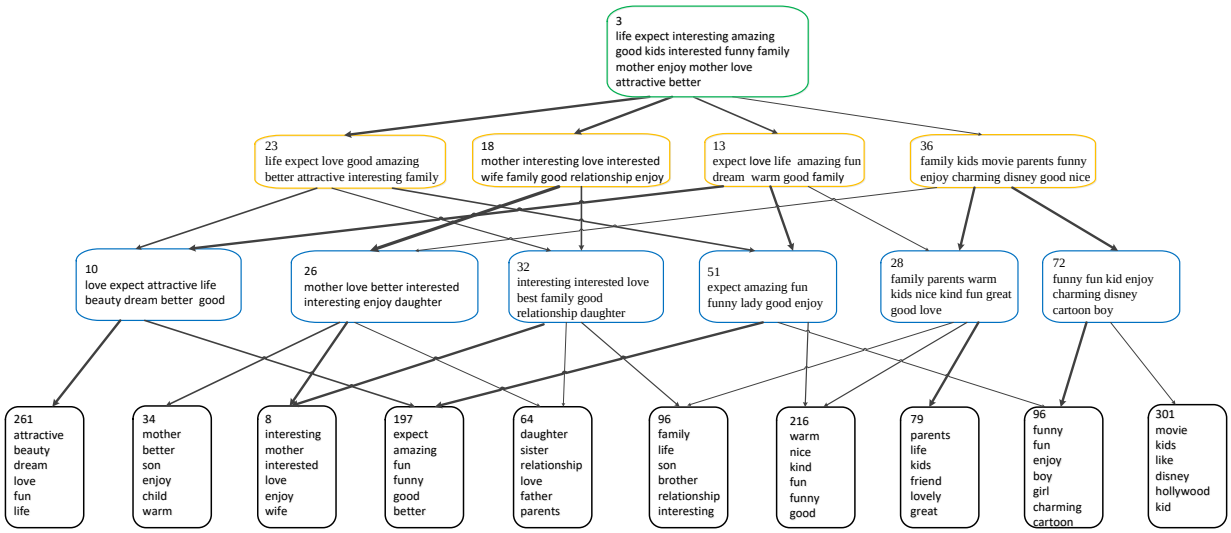


Fig. 12: Topics by supervised learning.

[9] M. D. Hoffman, D. M. Blei, C. Wang, and J. W. Paisley, "Stochastic variational inference." *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[10] N. Srivastava, R. Salakhutdinov, and G. E. Hinton, "Modeling documents with a deep Boltzmann machine," *Uncertainty in Artificial Intelligence*, pp. 616–624, 2013.

[11] R. Salakhutdinov and G. E. Hinton, "Replicated softmax: An undirected topic model," in *Advances in Neural Information Processing Systems*, 2009, pp. 1607–1614.

[12] A. Mnih and K. Gregor, "Neural variational inference and learning in belief networks," in *International Conference on Machine Learning*, 2014, pp. 1791–1799.

[13] Z. Gan, C. Chen, R. Henao, D. Carlson, and L. Carin, "Scalable deep Poisson factor analysis for topic modeling," in *International Conference on Machine Learning*, 2015, pp. 1823–1832.

[14] M. Zhou, L. Hannah, D. B. Dunson, and L. Carin, "Beta-negative binomial process and Poisson factor analysis," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 1462–1471.

[15] T. L. Griffiths, M. I. Jordan, J. B. Tenenbaum, and D. M. Blei, "Hierarchical topic models and the nested chinese restaurant process," in *Advances in neural information processing systems*, 2004, pp. 17–24.

[16] R. Ranganath, L. Tang, L. Charlin, and D. M. Blei, "Deep exponential families," in *International Conference on Artificial Intelligence and Statistics*, 2014, pp. 762–771.

[17] M. Zhou, Y. Cong, and B. Chen, "The Poisson gamma belief network," in *Advances in Neural Information Processing Systems*, 2015, pp. 3043–3051.

[18] ——, "Augmentable gamma belief networks," *Journal of Machine Learning Research*, vol. 17, no. 163, pp. 1–44, 2016.

[19] Y. A. Ma, T. Chen, and E. B. Fox, "A complete recipe for stochastic gradient MCMC," in *Advances in Neural Information Processing Systems*, 2015, pp. 2917–2925.

[20] S. Patterson and Y. W. Teh, "Stochastic gradient Riemannian Langevin dynamics on the probability simplex," in *Advances in Neural Information Processing Systems*, 2013, pp. 3102–3110.

[21] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *International Conference on Machine Learning*, 2011, pp. 681–688.

[22] Y. Cong, B. Chen, H. Liu, and M. Zhou, "Deep latent Dirichlet allocation with topic-layer-adaptive stochastic gradient Riemannian MCMC," in *International Conference on Machine Learning*, 2017.

[23] J. R. Foulds, L. Boyles, C. Dubois, P. Smyth, and M. Welling, "Stochastic collapsed variational Bayesian inference for latent Dirichlet allocation," in *Knowledge Discovery and Data Mining*, 2013, pp. 446–454.

[24] R. Ranganath, S. M. Gerrish, and D. M. Blei, "Black box variational inference," in *International Conference on Artificial Intelligence and Statistics*, 2013, pp. 814–822.

[25] T. Chen, E. B. Fox, and C. Guestrin, "Stochastic gradient Hamiltonian

Monte Carlo," in *International Conference on Machine Learning*, 2014, pp. 1683–1691.
[26] N. Ding, Y. Fang, R. Babbush, C. Chen, R. D. Skeel, and H. Neven, "Bayesian sampling using stochastic gradient Thermostats," in *Advances in Neural Information Processing Systems*.
[27] D. P. Kingma and M. Welling, "Stochastic gradient VB and the variational auto-encoder," in *International Conference on Learning Representations*, 2014.
[28] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International Conference on Machine Learning*, 2014, pp. 1278–1286.
[29] C. K. Sonderby, T. Raiko, L. Maaloe, S. K. Sonderby, and O. Winther, "Ladder variational autoencoders," in *Advances in Neural Information Processing Systems*, 2016, pp. 3738–3746.
[30] Z. Dai, A. C. Damianou, J. Gonzalez, and N. D. Lawrence, "Variational auto-encoded deep Gaussian processes," in *International Conference on Learning Representations*, 2016.
[31] G. Ishaan, K. Kundan, A. Faruk, T. Adrien, Ali, V. Francesco, V. David, and C. Aaron, "Pixelvae: A latent variable model for natural images," in *International Conference on Learning Representations*, 2017.
[32] A. Srivastava and C. Sutton, "Autoencoding variational inference for topic models," in *International Conference on Learning Representations*, 2017.
[33] D. A. Knowles, "Stochastic gradient variational Bayes for gamma approximating distributions," *arXiv preprint arXiv:1509.01631*, 2015.
[34] F. J. Ruiz, M. K. Titsias, and D. M. Blei, "The generalized reparameterization gradient," in *Advances in Neural Information Processing Systems*, 2016, pp. 460–468.
[35] C. Naesseth, F. Ruiz, S. Linderman, and D. Blei, "Reparameterization gradients through acceptance-rejection sampling algorithms," in *International Conference on Artificial Intelligence and Statistics*, 2017, pp. 489–498.
[36] Y. Ma, T. Chen, and E. Fox, "A complete recipe for stochastic gradient MCMC," in *Advances in Neural Information Processing Systems*, 2015, pp. 2899–2907.
[37] C. Lee and J. Chien, "Deep unfolding inference for supervised topic model," in *International Conference on Acoustics, Speech, and Signal Processing*, 2016, pp. 2279–2283.
[38] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in *Advances in Neural Information Processing Systems*, 2008, pp. 121–128.
[39] S. Lacostejulien, F. Sha, and M. I. Jordan, "DiscLDA: Discriminative learning for dimensionality reduction and classification," in *Advances in Neural Information Processing Systems*, 2009, pp. 897–904.
[40] H. Zhang, B. Chen, D. Guo, and M. Zhou, "Whai: Weibull hybrid autoencoding inference for deep topic modeling," in *international conference on learning representations*, 2018.
[41] M. Zhou, O. Padilla, and J. G. Scott, "Priors for random count matrices derived from a family of negative binomial processes," *J. Amer. Statist. Assoc.*, vol. 111, no. 515, pp. 1144–1156, 2016.
[42] G. Golub and C. V. Loan, "Matrix computations (3rd edition)," *JHU Press*.
[43] Y. Cong, B. Chen, and M. Zhou, "Fast simulation of hyperplane-truncated multivariate normal distributions," *Bayesian Analysis*, 2017.
[44] G. Polatkan, M. Zhou, L. Carin, D. M. Blei, and I. Daubechies, "A Bayesian nonparametric approach to image super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 346–358, 2015.
[45] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.
[46] D. M. Blei, M. I. Jordan, and J. W. Paisley, "Variational Bayesian inference with stochastic search," in *International Conference on Machine Learning*, 2012, pp. 1363–1370.
[47] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.
[48] M. Zhou, "Parsimonious Bayesian deep networks," in *Advances in Neural Information Processing Systems*, 2018.
[49] G. Marsaglia and W. W. Tsang, "A simple method for generating gamma variables," *ACM Transactions on Mathematical Software*, vol. 26, no. 3, pp. 363–372, 2000.
[50] S. Mandt, M. D. Hoffman, and D. M. Blei, "Stochastic gradient descent as approximate Bayesian inference," *Journal of Machine Learning Research*, 2017.
[51] C. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," *International Conference on Artificial Intelligence and Statistics*, pp. 562–570, 2014.
[52] A. Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems*, 2011, pp. 2348–2356.
[53] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *International Conference on Machine Learning*, 2015.
[54] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.
[55] R. Henao, Z. Gan, J. T. Lu, and L. Carin, "Deep Poisson factor modeling," in *Advances in Neural Information Processing Systems*, 2015, pp. 2800–2808.
[56] M. Hoffman, D. Blei, and F. Bach, "Online learning for latent Dirichlet allocation," in *Advances in Neural Information Processing Systems*, 2010, pp. 856–864.
[57] H. Larochelle and S. Lauly, "A neural autoregressive topic model," in *Advances in Neural Information Processing Systems*, 2012, pp. 2708–2716.
[58] J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan, "Nested hierarchical Dirichlet processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 256–270, 2015.
[59] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, "Evaluation methods for topic models," in *International Conference on Machine Learning*, 2009, pp. 1105–1112.
[60] J. Paisley, C. Wang, and D. Blei, "The discrete infinite logistic normal distribution for mixed-membership modeling," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 74–82.
[61] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. C. Courville, "Adversarially learned inference," in *International Conference on Learning Representations*, 2017.
[62] C. Li, J. Zhu, T. Shi, and B. Zhang, "Max-margin deep generative models," in *Advances in Neural Information Processing Systems*, 2015, pp. 1837–1845.
[63] M. Alireza and F. Brendan, "Adversarial autoencoders," in *International Conference on Learning Representations workshop*, 2016.
[64] C. Li, C. Chen, D. E. Carlson, and L. Carin, "Preconditioned stochastic gradient Langevin dynamics for deep neural networks," in *Conference on Artificial Intelligence*, 2016, pp. 1788–1794.
[65] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *ACL: Meeting of the Association for Computational Linguistics*, 2011, pp. 142–150.
[66] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," *north american chapter of the association for computational linguistics*, pp. 103–112, 2015.
[67] ——, "Supervised and semi-supervised text categorization using lstm for region embeddings," 2016, pp. 526–534.
[68] J. Zhu, N. Chen, H. Perkins, and B. Zhang, "Gibbs max-margin topic models with data augmentation," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1073–1110, 2014.
[69] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Advances in Neural Information Processing Systems*, 2015, pp. 3079–3087.

APPENDIX A
NAIVE DERIVATION OF THE FISHER INFORMATION MATRIX OF PGBN

We have discussed that it is difficult to calculate the FIM for PGBN expressed in (1) because of the coupled relationships between layers. In this section, we provide more detailed discussions.

For simplicity, we take for example a two-layer PGBN, expressed as

$$\begin{aligned}\boldsymbol{\theta}_n^{(2)} &\sim \text{Gam}\left(\boldsymbol{r}, 1/c_n^{(3)}\right),\\ \boldsymbol{\theta}_n^{(1)} &\sim \text{Gam}\left(\boldsymbol{\Phi}^{(2)}\boldsymbol{\theta}_n^{(2)}, 1/c_n^{(2)}\right),\\ \boldsymbol{x}_n &\sim \text{Pois}\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_n^{(1)}\right),\end{aligned} \tag{38}$$

and focus on a specific element $\boldsymbol{\Phi}_{vk}^{(2)}$ only. With the FIM definition

$$G(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{\Pi}|\boldsymbol{z}}\left[-\frac{\partial^2}{\partial \boldsymbol{z}^2}\ln p(\boldsymbol{\Pi}|\boldsymbol{z})\right], \tag{39}$$

where $\boldsymbol{\Pi}$ denotes the set of all observed and local variables, and $\boldsymbol{z}$ denotes the set of all global variables, one may show that the $\boldsymbol{\Phi}^{(2)}$-relevant part in $\ln p(\boldsymbol{\Pi}|\boldsymbol{z})$ is

$$\sum_{vn}\left[\boldsymbol{\Phi}_{v:}^{(2)}\boldsymbol{\theta}_{:n}^{(2)}\ln\left(c_n^{(2)}\boldsymbol{\theta}_{vn}^{(1)}\right) - \ln\Gamma\left(\boldsymbol{\Phi}_{v:}^{(2)}\boldsymbol{\theta}_{:n}^{(2)}\right)\right]. \tag{40}$$

Accordingly, with $\psi'(\cdot)$ denoted as the the trigamma function, for $\boldsymbol{\Phi}_{vk}^{(2)}$ we have

$$\mathbb{E}\left[-\frac{\partial^2}{\partial[\boldsymbol{\Phi}_{vk}^{(2)}]^2}\ln p(\boldsymbol{\Pi}|\boldsymbol{z})\right] = \mathbb{E}\left[\sum_n \psi'\left(\boldsymbol{\Phi}_{v:}^{(2)}\boldsymbol{\theta}_{:n}^{(2)}\right)\left[\boldsymbol{\theta}_{:n}^{(2)}\right]^2\right], \tag{41}$$

which is an expectation that is difficult to evaluate.

APPENDIX B
PROOF OF DLDA EXPRESSION

Note that the counts in $x_{vn}^{(l)} \sim \text{Pois}(q_j^{(l)}\sum_{k=1}^K \phi_{vk}^{(l)}\theta_{kn}^{(l)})$ can be augmented as

$$\begin{aligned}x_{vn}^{(l)} &= \sum_{k=1}^K x_{vkn}^{(l)},\\ x_{vkn}^{(l)} &\sim \text{Pois}\left(q_j^{(l)}\phi_{vk}^{(l)}\theta_{kn}^{(l)}\right),\end{aligned} \tag{42}$$

which, according to Lemma 4.1 of [14], can be equivalently expressed as

$$\begin{aligned}\left(x_{vkn}^{(l)}\right)_v &\sim \text{Mult}\left(m_{kn}^{(l)(l+1)}, \boldsymbol{\phi}_k^{(l)}\right),\\ m_{kn}^{(l)(l+1)} &\sim \text{Pois}\left(q_j^{(l)}\theta_{kn}^{(l)}\right),\end{aligned} \tag{43}$$

where $m_{kn}^{(l)(l+1)} := \sum_{v=1}^{K_{l-1}} x_{vkn}^{(l)}$. Marginalizing out $\theta_{vn}^{(l)} \sim \text{Gam}\left(\sum_{k=1}^{K_{l+1}}\boldsymbol{\phi}_{vk}^{(l+1)}\boldsymbol{\theta}_{kn}^{(l+1)}, 1/c_n^{(l+1)}\right)$ from (43) leads to

$$m_{vj}^{(l)(l+1)} \sim \text{NB}\left(\sum_{k=1}^{K_{l+1}}\boldsymbol{\phi}_{vk}^{(l+1)}\boldsymbol{\theta}_{kn}^{(l+1)}, p_j^{(l+1)}\right), \tag{44}$$

which can be augmented as

$$\begin{aligned}m_{vj}^{(l)(l+1)} &\sim \text{SumLog}\left(x_{vn}^{(l+1)}, p_n^{(l+1)}\right),\\ x_{vj}^{(l+1)} &\sim \text{Pois}\left(q_n^{(l+1)}\sum_{k=1}^{K_{l+1}}\boldsymbol{\phi}_{vk}^{(l+1)}\boldsymbol{\theta}_{kn}^{(l+1)}\right).\end{aligned} \tag{45}$$

APPENDIX C
DERIVATION OF THE $\Gamma(\cdot)$ FUNCTIONS IN SG-MCMC

With $\mathbf{D}(\boldsymbol{z}) = \mathbf{G}(\boldsymbol{z})^{-1}$, $\mathbf{Q}(\boldsymbol{z}) = \mathbf{0}$, and the block-diagonal Fisher information matrix (FIM) $\mathbf{G}(\boldsymbol{z})$ in (6), it is straightforward to show that $\frac{\partial}{\partial \boldsymbol{\varphi}_k}\left[\mathbf{D}(\boldsymbol{z}) + \mathbf{Q}(\boldsymbol{z})\right]$ is non-zero only in the $\boldsymbol{\varphi}_k$-related block $\mathbf{I}(\boldsymbol{\varphi}_k)$ in (7). Therefore, we focus on this block and have

$$\Gamma_v(\boldsymbol{\varphi}_k) = \sum_u \frac{\partial}{\partial \varphi_{uk}}\left[\mathbf{I}^{-1}_{vu}(\boldsymbol{\varphi}_k)\right], \tag{46}$$

where $\mathbf{I}^{-1}_{vu}(\boldsymbol{\varphi}_k) = M_k^{-1}\left[\text{diag}(\boldsymbol{\varphi}) - \boldsymbol{\varphi}\boldsymbol{\varphi}^T\right]$. Accordingly, we have

$$\begin{aligned}\Gamma_v(\boldsymbol{\varphi}_k) =& M_k^{-1}\sum_u \frac{\partial}{\partial \varphi_{uk}}[\delta_{u=v}\varphi_{uk} - \varphi_{vk}\varphi_{uk}] \\ =& M_k^{-1}(1 - V\varphi_{vk}).\end{aligned} \tag{47}$$

Since $\mathbf{G}(\boldsymbol{z})$ is a block-diagonal with its $\boldsymbol{r}$-relevant block being $\mathbf{I}(\boldsymbol{r}) = M^{(L+1)}\text{diag}(1/\boldsymbol{r})$, according to the definition of $\Gamma(\cdot)$, it is straightforward to show that

$$\begin{aligned}\Gamma_k(\boldsymbol{r}) &= \sum_u \frac{\partial}{\partial r_u}\left[\mathbf{I}^{-1}_{ku}(\boldsymbol{r})\right] \\ &= \sum_u \frac{\partial}{\partial r_u}\left[\delta_{u=k}\frac{r_u}{M^{(L+1)}}\right] \\ &= 1/M^{(L+1)}.\end{aligned} \tag{48}$$

APPENDIX D
HIERARCHICAL TOPICS LEARNED FROM WIKI

In this section, we present more examples of the hierarchical topics learned from the Wiki dataset in Figs. 13 and 14, where Fig. 1 shows a subnetwork from the same topic tree as Fig. 5 in the main manuscript does, while Fig. 2 is obtained based on a four-layer DATM-WHAI.

The semantic meaning of each topic and the connections between different topics are highly interpretable. For example, the subnetwork in Fig. 2 mainly talks about "war" and "government," shown as Topic 6, which is further divided into two subtopics focusing on "public/law" and "military/battle," respectively.

Moreover, comparing Fig. 1 with Fig. 5 in the main manuscript, although the same or similar words might appear in different topics belonging to the same subnetwork, the differences between the two subnetworks are evident, which demonstrates that different subnetworks could capture distinct semantics existing in the corpus.

Comparing with Fig. 5 in the main manuscript, the higher-layer topics of these two additional example subnetworks exhibit more distinct meanings. For example, in Fig. 14, Topics 5 and 28 at layer 3 talk about "government/public/law" and "government/war/army." Similar results can also be found in Fig. 15 and Fig. 18. Moreover, although some similar or same words are in the different topics belonging to the same subnetwork, examining Fig. 5 in the main manuscript and Figs. 13 and 14, for the same dataset (WIKI), the different subnetworks are evidently different from each other.

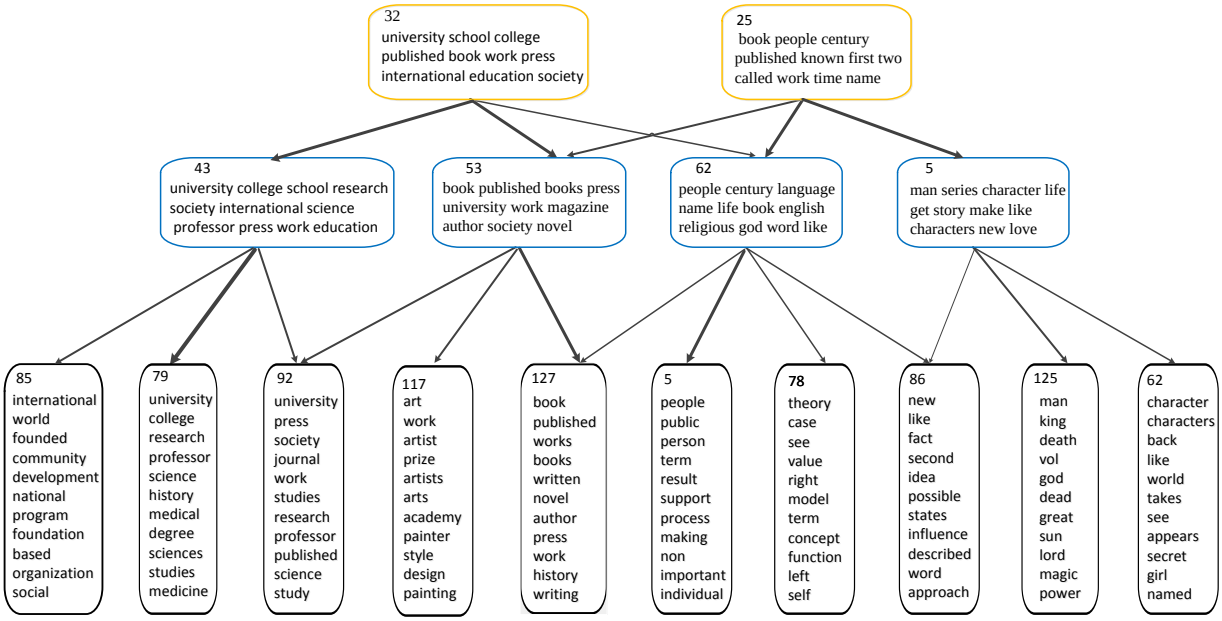


Fig. 13: Example of hierarchical topics learned from Wiki by a three-layer DATM-WHAI of size 128-64-32

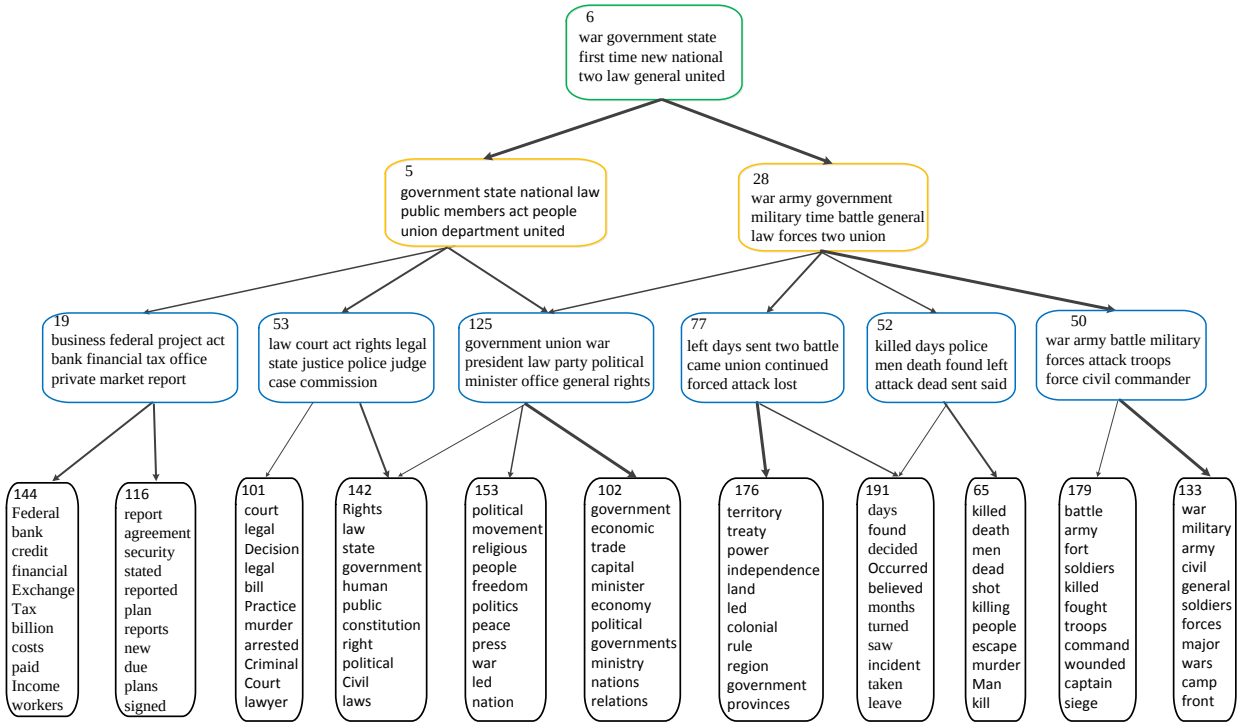


Fig. 14: Example of hierarchical topics learned from Wiki by a four-layer DATM-WHAI of size 256-128-64-32

APPENDIX E
HIERARCHICAL TOPICS LEARNED FROM 20NEWS AND NIPS12 BY DATM, HLDA AND DEF

In order to better understand the distinction of DATM in learning hierarchical topics, we compare the results of DATM with hLDA [15] and DEF [16]; we refer to the original publications on how their topics are visualized. Different from DATM and DEF, hLDA arranges its topics into a tree-structured L-level hierarchy and a document can only choose a mixture of L topics along a document-specific root-to-leaf path. This construction makes the topics of hLDA closer to the root node to contain more commonly used words, as shown in Fig. 4.

On the contrary, in both DATM and DEF, a document is not restricted to choose the topics in a single path of the inferred topic hierarchy. DATM relates the topic weights of adjacent layers via the gamma shape parameters, as discussed in Eq. (2) in the main manuscript, whereas DEF does so via the gamma scale parameters. As shown in Figs. 15 and 18 for DATM and Figs. 17 and 20 for DEF, the topics at the first layer of both models share similarities and their higher-layer topics are the weighted combinations of the lower-layer ones. However, it is straightforward for DATM to visualize its topics at higher layers. By contrast, as in Ranganath et al. [16], the higher-layer topics of DEF need to be manually summarized.

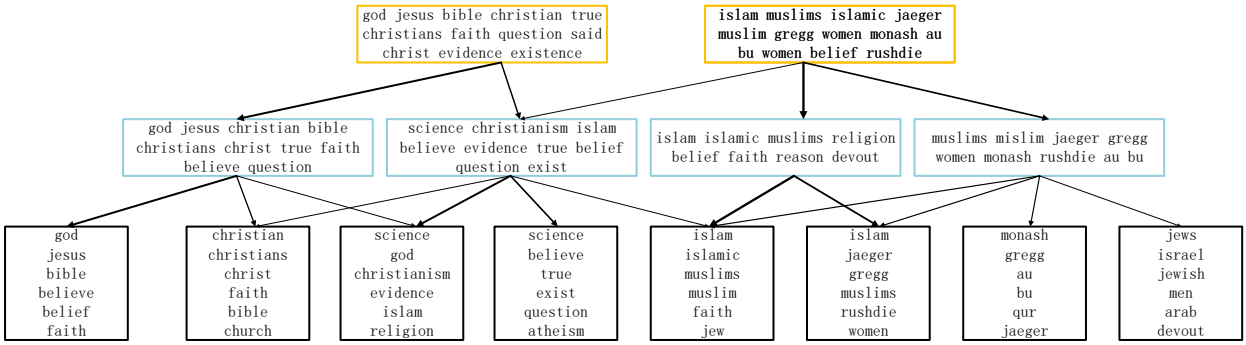


Fig. 15: Example of hierarchical topics learned from 20News by a three-hidden-layer DATM-WHAI.

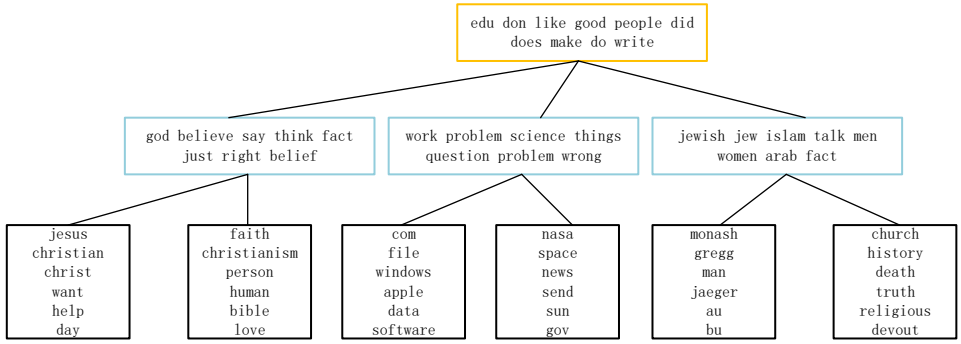


Fig. 16: Example of hierarchical topics learned from 20News by a three-hidden-layer hLDA.

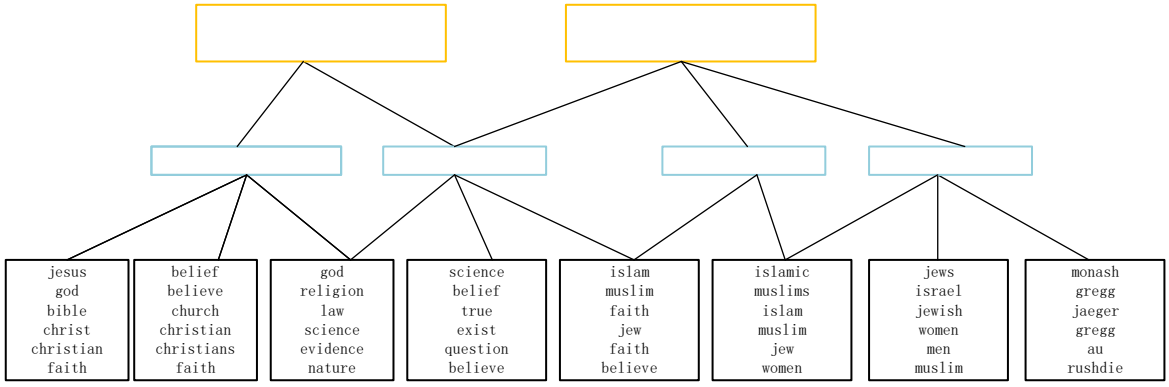


Fig. 17: Example of hierarchical topics learned from 20News by a three-hidden-layer DEF. Such visualization follows Ranganath et al. [16], where the high-level topics are vacant.

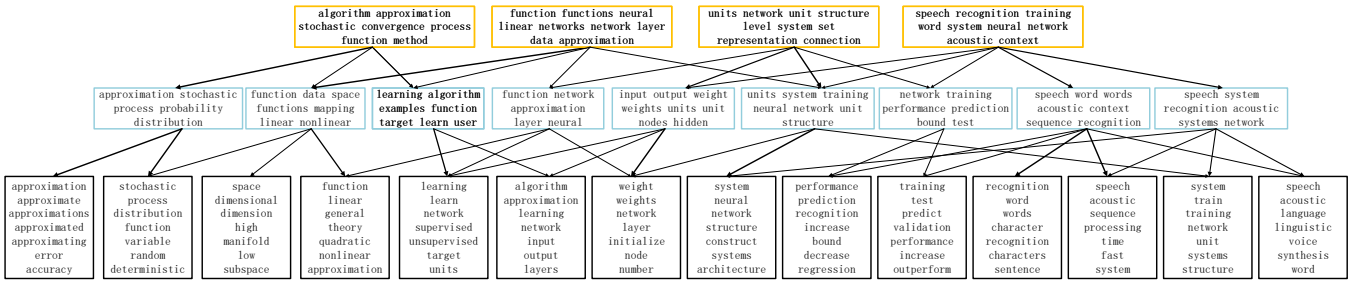


Fig. 18: Example of hierarchical topics learned from NIPS12 by a three-hidden-layer DATM-WHAI.

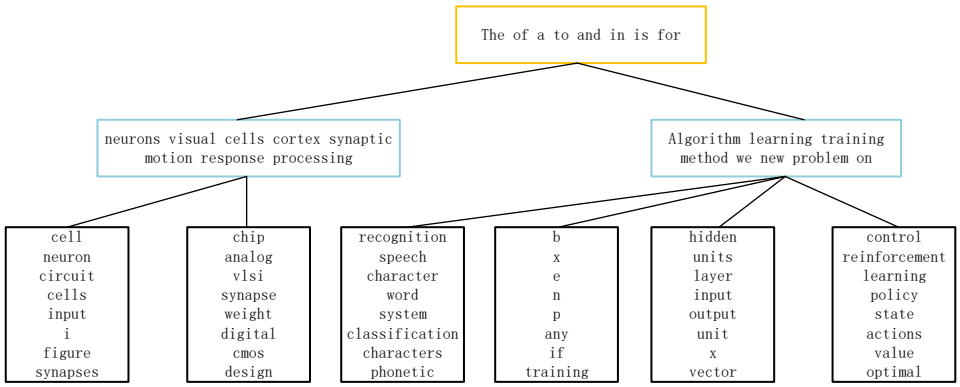


Fig. 19: Example of hierarchical topics learned from NIPS12 by a three-hidden-layer hLDA (from the original paper).

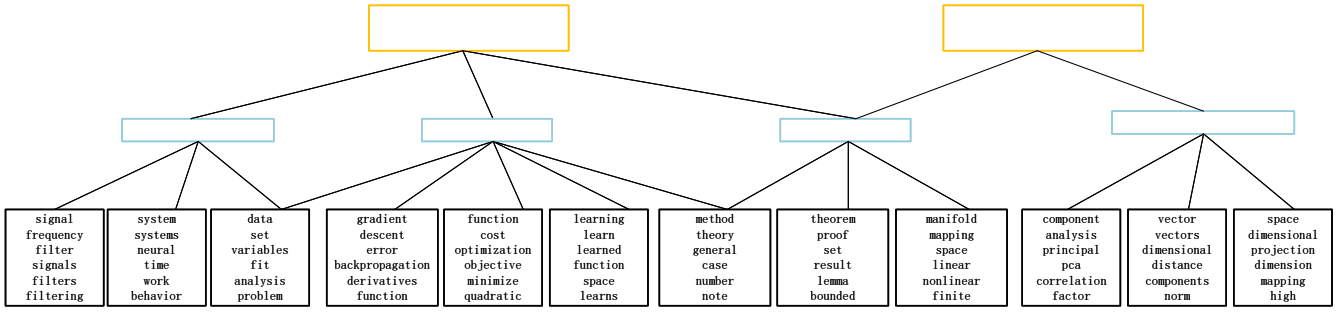


Fig. 20: Example of hierarchical topics learned from NIPS12 by a three-hidden-layer DEF. Such visualization follows Ranganath et al. [16], where the high-level topics are vacant.

APPENDIX F
MANIFOLD ON DOCUMENTS

From a sci.medicine document to an eci.space one

1. com, writes, article, edu, medical, pitt, pain, blood, disease, doctor, medicine, treatment, patients, health, ibm
2. com, writes, article, edu, space, medical, pitt, pain, blood, disease, doctor, data, treatment, patients, health
3. space, com, writes, article, edu, data, medical, launch, earth, states, blood, moon, disease, satellite, medicine,
4. space, data, com, writes, article, edu, launch, earth, states, moon, satellite, shuttle, nasa, price, lunar
5. space, data, launch, earth, states, moon, satellite, case, com, shuttle, price, nasa, price, lunar, writes,
6. space, data, launch, earth, states, moon, orbit, satellite, case, shuttle, price, nasa, system, lunar, spacecraft

From a alt.atheism document to a soc.religion.christian one

1. god, just, want, moral, believe, religion, atheists, atheism, christian, make, atheist, good, say, bible, faith
2. god, just, want, believe, jesus, christian, atheists, bible, atheism faith, say, make, religious, christians, atheist
3. god, jesus, just, faith, believe, christian, bible, want, church, say, religion, moral, lord, world, writes
4. god, jesus, faith, just, bible, church, christ, believe, say, writes, lord, religion, world, want, sin
5. god, jesus, faith, church, christ, bible, christian, say, write, lord, believe, truth, world, human, holy
6. god, jesus, faith, church, christ, bible, writes, say, christian, lord, sin, human, father, spirit, truth

From a com.graphics document to a comp.sys.ibm.pc.hardware one

1. image, color, windows, files, image, thanks, jpeg, gif, card, bit, window, win, help, colors, format
2. image, windows, color, files, card, images, jpeg, thanks, gif, bit, window, win, colors, monitor, program
3. windows, image, color, card, files, gov, writes, nasa, article, images, program, jpeg, vidio, display, monitor
4. windows, gov, writes, nasa, article, card, going, program, image, color, memory, files, software, know, screen
5. gov, windows, writes, nasa, article, going, dos, card, memory, know, display, says, screen, work, ram
6. gov, writes, nasa, windows, article, going, dos, program, card, memory, software, says, ram, work, running