Appendix

A Inference Network of dc-ETM

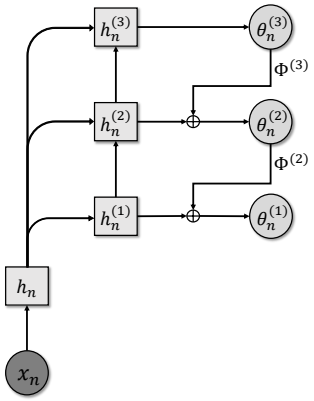

Figure 6: Overview of the hierarchical upward and downward inference network (encoder) of proposed dc-ETM.

As a usual VAE-like model, the inference network (encoder) in dc-ETM is designed to approximate the gamma distributed posteriors of latent representations of the corresponding generative model (decoder), specifically $p(\boldsymbol{\theta}_n|\boldsymbol{x}_n)$ defined in Eq. (1). As the gamma distribution, which can generate sparse and non-negative random variables, is not reparameterizable with respect to its shape parameter, we introduce the Weibull distribution [20] to construct our inference network due to its attractive properties as discussed in Appendix B, one of which is that a latent variable $x \sim \text{Weibull}(k, \lambda)$ can be easily reparameterized as

$$x = \lambda(-\ln(1-\varepsilon))^{1/k}, \;\; \varepsilon \sim \text{Uniform}(0, 1). \tag{17}$$

Specifically, following traditional VAEs, we first factorize the variational posterior distribution $q(\boldsymbol{\theta}_n|\boldsymbol{x}_n)$ defined by the inference network in a hierarchical manner as follows

$$q(\boldsymbol{\theta}_n|\boldsymbol{x}_n) = q(\boldsymbol{\theta}_n^{(L)}|\boldsymbol{x}_n) \prod_{l=1}^{L-1} q(\boldsymbol{\theta}_n^{(l)}|\boldsymbol{\theta}_n^{(l+1)}, \boldsymbol{x}_n), \tag{18}$$

which is expected to be flexible enough to well approximate the true posterior distribution $p(\boldsymbol{\theta}_n|\boldsymbol{x}_n)$. Then, for the design of the network structure, we develop a novel Weibull-based upward-downward inference network, which contains both bottom-up deterministic and top-down stochastic paths, contributing to reducing the noise in the procedure of inferencing stochastic latent variables that are higher in the hierarchy [16, 17].

As the inference network shown in the left part of 6, the bottom-up deterministic path takes the BoW vector $\boldsymbol{x}_n$ as input to obtain hierarchical deterministic latent representations $\{\boldsymbol{h}_n^{(l)}\}_{l=1}^L$ as follows:

$$\boldsymbol{h}_n^{(l)} = \text{MLP}(\boldsymbol{h}_n^{(l-1)} \oplus \boldsymbol{h}_n), \tag{19}$$

through specifically defining $\boldsymbol{h}_n = \text{MLP}(\boldsymbol{x}_n)$ and $\boldsymbol{h}_n^{(1)} = \text{MLP}(\boldsymbol{h}_n)$ with MLP indicating a two-layered fully connected network and $\oplus$ being the concatenation operation at feature dimension. Then, each deterministic latent representation $\boldsymbol{h}_n^{(l)}$ is further transferred into

$$\hat{\boldsymbol{k}}_n^{(l)} = \text{Relu}(\text{Linear}(\boldsymbol{h}_n^{(l)})), \tag{20}$$

$$\hat{\boldsymbol{\lambda}}_n^{(l)} = \text{Relu}(\text{Linear}(\boldsymbol{h}_n^{(l)})), \tag{21}$$

where Linear is a dense fully connected layer. Finally, taking $\boldsymbol{\Phi}^{(l+1)}\boldsymbol{\theta}_n^{(l+1)}$ as the prior information passing from the deeper layer, the variational posterior $q(\boldsymbol{\theta}_n^{(l)}|-)$ can be obtained with the stochastic

up-down path as

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

where Softplus function is applied to ensure positive Weibull shape and scale parameters; $\hat{\boldsymbol{k}}_n^{(l)}$ and $\hat{\boldsymbol{\lambda}}_n^{(l)}$ incorporate the information passing from $\boldsymbol{h}_n^{(l)}$; and $\boldsymbol{\theta}_n^{(l)}$ can be obtained with the reparameterization technique defined in Eq. (17).

We emphasize that the inference network of our dc-ETM can be naturally treated as an inverse procedure of the generation process of $\boldsymbol{x}_n$, whose generation is conditional on the latent representations $\{\boldsymbol{\theta}_n^{(l)}\}_{l=1}^L$ across all hidden layers. Inverse to the generative network, the inference network can directly inject the data information into the latent representation of each layer through $\boldsymbol{h}_n$ and allow all the stochastic latent variables $\{\boldsymbol{\theta}_n^{(l)}\}_{l=1}^L$ to have a deterministic dependency on the observation $\boldsymbol{x}_n$, empirically alleviating top stochastic latent variables from being collapsed [17].

B Properties of Weibull distribution

• Similar density characteristics with Gamma Distribution

The Weibull distribution has similar characteristics with gamma distribution, *i.e.*, the density functions of the two distributions are quite similar

$$
\begin{aligned}
&\text{Weibull PDF: } P(x|k,\lambda) = \frac{k}{\lambda^k} x^{k-1} e^{(x/\lambda)^k}, \\
&\text{Gamma PDF: } P(x|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}.
\end{aligned} \tag{23}
$$

• Easily Reparameterization

The latent variable $x \sim \text{Weibull}(k, \lambda)$ can be reparameterized as

$$
x = \lambda(-\ln(1-\varepsilon))^{1/k}, \;\; \varepsilon \sim \text{Uniform}(0, 1), \tag{24}
$$

leading to an expedient and numerically stable gradient calculation.

• Analytic KL-Divergence

The KL-divergence between the Weibull and gamma distributions has an analytic expression formulated as

$$
\begin{aligned}
&\text{KL}(\text{Weibull}(k,\lambda)||\text{Gamma}(\alpha,\beta)) = -\alpha\ln\lambda + \frac{\gamma\alpha}{k} \\
&+ \ln k + \beta\lambda\Gamma(1+\frac{1}{k}) - \gamma - 1 - \alpha\ln\beta + \ln\Gamma(\alpha).
\end{aligned} \tag{25}
$$

C PG-based Training Algorithm

Modeling the generation of long sequence with RL-based methods has achieved great success, but still suffers from high variance during model training. The common ways to conduct policy gradient method for training include: **i)** employ the score-ratio gradient approximator like REINFORCE [46] to backward the gradients from Q-value $Q^\pi(s_n^{(l)}, a_n^{(l)})$ to the parameters of the policy network $\boldsymbol{\pi}$; **ii)** introduce an additional "critic" network to estimate the Q-value $Q^\pi(s_n^{(l)}, a_n^{(l)})$ as $\hat{Q}^\pi(s_n^{(l)}, a_n^{(l)})$, then backward the gradients from estimated Q-value $\hat{Q}^\pi(s_n^{(l)}, a_n^{(l)})$ to the parameters of the policy network $\boldsymbol{\pi}$ through methods like deep deterministic policy gradient (DDPG) [34]. We emphasize that, both methods above are developed for the RL applications where the Q-value function is not differentiable, e.g., the reward $r(s_n^{(l)}, a_n^{(l)})$ at every time step l is given by the non-differentiable black-box environment simulator, or for the applications with hundreds or even thousands time steps, where the direct

computation of the gradient from $Q^{\pi}(s_n^{(l)}, a_n^{(l)}) = r(s_n^{(l)}, a_n^{(l)}) + \mathbb{E}_{\pi}\left[\sum_{i=1}^{l-1} \gamma^i r(s_n^{(l-i)}, a_n^{(l-i)})\right]$ to $\boldsymbol{\pi}$ is very expensive. However, the REINFORCE estimator is notorious for its high variance, which would cause the learning of long sequence hard to converge, and the introducing of the "critic" network would also bring extra bias for the gradients and computation burden.

Moving beyond the gradient estimator of REINFORCE method or introducing the critic network, we design a more stable way to model the sequential generative process in deep topic model here. Specifically, the number of time steps here is the number of layers L of the topic model, e.g., L is 5 in this paper, which is a much smaller number compared to the hundreds of time step in other RL applications. Besides, we design the $r(s_n^{(l)}, a_n^{(l)})$ as a totally differentiable function in Eq. 14, which leads to a differentiable Q-value function $Q^{\pi}(s_n^{(l)}, a_n^{(l)})$. More specifically, recall the purpose of policy gradient method is to maximize the Q-value by applying gradient descent to the policy model $\boldsymbol{\pi}$, we develop the Policy Gradient based variational inference algorithm for dc-ETM in Algorithm 1.

Algorithm 1 The Policy Gradient based variational inference algorithm for dc-ETM.

Set minibatch size m and the number of Layer L;
Initialize the encoder parameters $\boldsymbol{\Omega}$ and decoder parameters $\boldsymbol{\Psi}$;
for $iter = 1, 2, \cdots$ **do**
  Randomly select a minibatch of m documents to form a subset $\boldsymbol{X} = \{\boldsymbol{x}_j\}_{1,m}$;
  Draw random noise $\{\boldsymbol{\epsilon}_i^l\}_{i=1,l=1}^{m,L}$ from uniform distribution;
  for Layer $l = L, L-1, \cdots, 1$ **do**
    for $i = l, l-1, \cdots, 1$ **do**
      Calculate $r(s_n^{(i)}, a_n^{(i)})$ according to Eq. (14);
    end for
    Calculate $-\nabla_{\boldsymbol{\Omega},\boldsymbol{\Psi}} Q^{\pi}(s_n^{(l)}, a_n^{(l)}; \boldsymbol{X}, \{\boldsymbol{\epsilon}_i^l\}_{i=1,l=1}^{m,L})$ according to Eq. (15), and update $\boldsymbol{\Omega}, \boldsymbol{\Psi}$ jointly;
  end for
end for

D Datasets

R8 is a subset of 7,674 documents selected form 8 different review groups of the Reuters 21578 dataset, and has been split into a training set of 5,485 documents and a testing set of 2,189 ones. **20News** dataset consists of 18,774 documents from 20 various new groups and has been split into a training set of 11,314 documents and a testing set of 7,532 ones. **RCV1** is an archive of 804,114 manually categorized newswire stories made available by Reuters. **WebKB** is a dataset that includes web pages from computer science departments of various universities, which has 4,518 web pages that are categorized into 6 imbalanced categories.

E Error Bars

We randomly run 5 seeds for our method in experiments and report the error bar as below.

Table 3: Error bar for our method in the comparisons of the average of perplexities and topic diversities across all hidden layers on various benchmarks.

Model	Perplexity			Topic Diversity		
	R8	20News	RCV1	R8	20News	RCV1
dc-ETM-α	521±6	730±7	912±11	0.212±0.002	0.281±0.001	0.435±0.003
dc-ETM-β	427±5	710±5	873±10	0.346±0.003	0.429±0.003	0.566±0.005
dc-ETM-α (Policy)	463±4	707±5	896±5	0.279±0.002	0.385±0.002	0.519±0.004
dc-ETM-β (Policy)	420±3	647±4	841±5	0.379±0.004	0.456±0.003	0.584±0.004

Table 4: Error bar of our method in the document clustering comparison on the 1st hidden layer or the concatenation of all hidden layers of different topic models.

Model	Layer	WebKB		20News		R8	
		Purity	NMI	Purity	NMI	Purity	NMI
dc-ETM-α	1	61.14±0.5	26.29±0.2	32.81±0.2	43.64±0.4	75.60±0.5	39.83±0.2
	All	63.18±0.4	28.35±0.2	41.83±0.4	44.52±0.3	76.31±0.5	43.73±0.4
dc-ETM-β	1	54.71±0.5	21.43±0.1	39.80±0.2	44.30±0.3	74.30±0.6	38.63±0.3
	All	67.29±0.4	33.60±0.3	45.00±0.2	46.20±0.3	76.25±0.5	45.64±0.4
dc-ETM-α (Policy)	1	49.71±0.3	14.86±0.1	37.88±0.2	43.56±0.2	71.65±0.4	32.73±0.2
	All	64.32±0.4	33.65±0.1	42.21±0.2	45.59±0.3	77.46±0.6	44.60±0.3
dc-ETM-β (Policy)	1	57.32±0.5	26.05±0.1	40.11±0.2	44.12±0.3	71.30±0.4	38.34±0.2
	All	69.32±0.5	38.53±0.3	48.60±0.4	55.79±0.4	78.29±0.6	48.62±0.5

F Comparison of topic quality

To make an intuitive comparison on the aspect of topic quality, we visualize the 5-layer topics learned by dc-ETM and Sawtooth on 20News dataset as shown in . Obviously, the topics learned by Sawtooth are quite similar, which potentially explains the reason why concatenating its hierarchical latent document representations cannot improve and even hurt the performance on downstream tasks. On the contrary, the developed dc-ETM can learn meaningful and diverse topics in higher layers, indicating that more data information is passed to higher layers to alleviate the phenomenon of collapse.

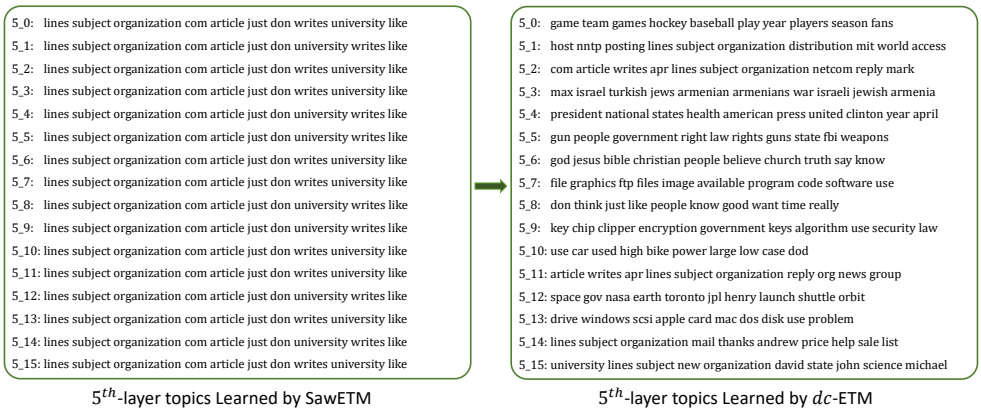

Figure 7: The 5-layer topics learned by dc-ETM and Sawtooth on 20News dataset, where each topic is interpreted by its top-10 words by sorting the word probabilities by descending order.

G Limitation

The limitation could be the computation burden brought by the extra $L-1$ training steps due to the Policy Gradient training method. However, since L is a small number, the extra computation burden is affordable. Besides, compared with the backbone GBN-based deep topic model, our method would not bring extra computation burden during the testing stage, but leads to performance improvements as shown in Section 5.

H Broader Impact

At first, we need to emphasize that this work is developed to mitigate the phenomenon of information decay in deep topic models, which has been widely disclosed in the topic modeling literature but few efforts have been made to address this challenge. The main difficulty is the need for carefully designing the probabilistic generative process to build the effective connection between the observation and the corresponding latent representations, on the premise of preserving the interpretable hierarchical topic modeling structure, rather than casually introducing skip-connections. Utilizing the natural hierarchy in deep topic models, we provide a general solution equipped with a novel perspective to incorporate RL-based training algorithm, which brings quite a few contributing ideas to this field.

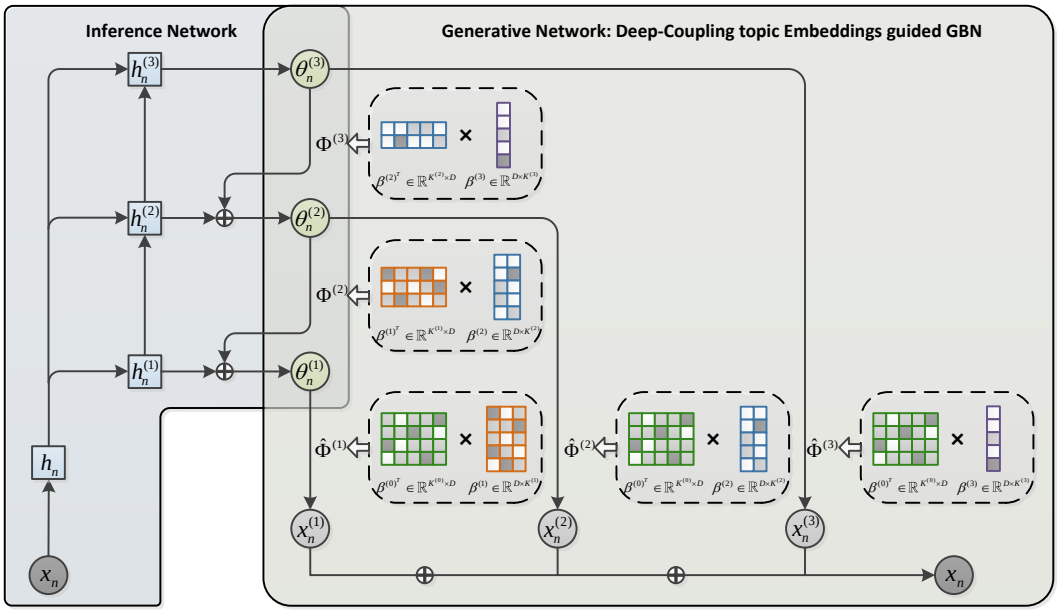

Figure 8: Overview of the proposed dc-ETM, where the left part is the hierarchical upward and downward inference network (encoder) and the right part is the generative network (decoder).

Please note that, we take an NTM named SawETM as an example in this paper to illustrate how we develop a dc-ETM, but dc-ETM can be also applied to extend NTMs with similar structures. Most of exsiting NTMs adopt VAE-liked structures, but none of them attempt to solve the "*posterior collapse*" phenomenon in essence, resulting in that the latent representations at higher layers exhibit similar or meaningless patterns (as shown in the Appendix F).

The significance of developing deep-coupling structure with RL-based training algorithm, which has effectively improved the quality of the latent representations of a deep topic model at higher layers, goes beyond a single specific model.

I The implementation details of dc-ETM variants

To have an intuitive understanding about the implementation details of the projection metrics in dc-ETM, we provide an overview of the network structure of a 3-layer dc-ETM in Fig. 8. As shown in Fig. 8, the topic matrix $\mathbf{\Phi}^{(l)}$ at layer l can be factorized into the product of two topic embedding matrices. Then, there are two kinds of choices to obtain the projection matrices, leading to the variants distinguished by the suffix $-\alpha$ and $-\beta$.

Specifically, for the variant dc-ETM-α defined in Eq. (2), the projection matrix $\hat{\mathbf{\Phi}}^{(3)}$ can be obtained by multiplying $\mathbf{\Phi}^{(1)}\mathbf{\Phi}^{(2)}\mathbf{\Phi}^{(3)}$, and then the augmented observation vector $\boldsymbol{x}_n^{(3)}$ can be generated from the Poisson distribution with a rate of $\alpha^{(3)}\hat{\mathbf{\Phi}}^{(3)}\boldsymbol{\theta}_n^{(3)}$. The other projection matrices $\hat{\mathbf{\Phi}}^{(2)}$ and $\hat{\mathbf{\Phi}}^{(1)}$ can be obtained in a simliar way, where specifically defining $\hat{\mathbf{\Phi}}^{(1)} := \mathbf{\Phi}^{(1)}$.

For the variant dc-ETM-β defined in Eq. (3), the projection matrix $\hat{\mathbf{\Phi}}^{(3)}$ can be obtained by directly multiplying $\boldsymbol{\beta}^{(0)T}$ and $\boldsymbol{\beta}^{(3)}$, which is more efficient than dc-ETM-α. Thus, in our consideration, with a shorter path for gradient propagation, the short connections in dc-ETM-β can perverse more data information than those in dc-ETM-α, resulting in more informative latent document representations to achieve better model performance with less "*posterior collapse*".

We emphasize that, the projection in dc-ETM-α could be applied to extend any existing deep topic models, while the projection in dc-ETM-β is only applicable for the NTMs equipped with topic embedding techniques.

J Preliminary of "*posterior collapse*"

J.1 Definition of "*posterior collapse*"

For a VAE-based model consisting of a decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ and an encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, the definition of posterior collapse is that the posterior of latent variables, denoted as $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, collapses to it prior $p_\theta(\boldsymbol{z})$, which is a non-informative distribution and independent of the data $\boldsymbol{x}$. It can be mathematically denoted as that the KL divergence between $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and $p_\theta(\boldsymbol{z})$ is close to zero, represented as $D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})] \approx 0$.

J.2 How to measure "*posterior collapse*"

As the definition of "*posterior collapse*", a promising metric to measure "*posterior collapse*" could be the KL-divergence between $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and $p_\theta(\boldsymbol{z})$, where a smaller KL-divergence score indicates that the posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ contains less data information and has a larger tendency to collapse to its non-informative prior. In Table 5, we compare our method dc-ETM-β(policy) (for brevity, we note it as "dc-ETM") with the previous SOTA NTM named SawETM with the metric of layer-wise KL divergence on 20News, R8, and RCV1 datasets. As the results shows, the KL divergence of SawETM gradually reduces to zero with the network going deeper, which indicates the occurrence of a serious degree of "*posterior collapse*" at higher layers and limited data information can be captured by these latent variables at higher layers. On the contrary, benefiting from the efficient skip connection between the observation $\boldsymbol{x}_n$ and multiple latent document representations $\{\boldsymbol{\theta}_n^{(l)}\}_{l=1}^L$, dc-ETM can achieve relatively larger KL-divergence scores at higher layers by alleviating "*posterior collapse*", leading to more expressive latent document representations for downstream tasks.

KL-divergence of each layer						
Metric	20News		R8		RCV1	
Layer #i	SawETM	dc-ETM	SawETM	dc-ETM	SawETM	dc-ETM
1	124	354	80.5	161	156	365
2	38.6	233	26.1	89.8	72.1	235
3	3.07	158	11.7	81.9	55.4	170
4	2.51	132	2.05	67.4	48.7	122
5	1.64	101	1.16	77.7	48.0	102

Table 5: Layer-wise KL divergence scores of SawETM and dc-ETM.

J.3 Why "*posterior collapse*" in a VAE-based model

Besides the experimental analysis, to have a theoretical understanding of the reason why *"posterior collapse"* widely exists in the higher layers of hierarchical VAEs, we try to explain the phenomenon of *"posterior collapse"* from the perspective of information theory. For ease of understanding, we use a vanilla L-layer hierarchical VAE with a top-down inference network as an example, where dc-ETMs can be all treated as VAE-based models. Specifically, following [18], we extend their theoretical explanation for *"posterior collapse"* in a single-layer VAE to a multi-layer version.

Separating the latent variables as the lower-level variables $\boldsymbol{z}_{\leq k} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_k\}$ and the higher-level ones $\boldsymbol{z}_{>k} = \{\boldsymbol{z}_{k+1}, ..., \boldsymbol{z}_L\}$, where $k \in \{0, ..., L-1\}$, then the ELBO for this hierarchical VAE can be reformulated as

$$\mathcal{L} = \mathbb{E}_{p(\boldsymbol{x})}\left[\mathbb{E}_{q_\phi(\boldsymbol{z}_{\leq k}|\boldsymbol{z}_{>k})q_\phi(\boldsymbol{z}_{>k}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z}_1)\right] - \sum_{l=1}^{L} D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_l|\boldsymbol{z}_{l+1})||p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{l+1}))\right], \quad (26)$$

where $q_\phi(\boldsymbol{z}_L|\boldsymbol{z}_{L+1}) := q_\phi(\boldsymbol{z}_L|\boldsymbol{x})$, $p_\theta(\boldsymbol{z}_L|\boldsymbol{z}_{L+1}) := p_\theta(\boldsymbol{z}_L)$, and the main contribution to the expected log-likelihood term is coming from the lower-level latent variables $\boldsymbol{z}_{\leq k}$ before the k-th hidden layer [17]. Once the generation capacity of the generative model $p_\theta(\boldsymbol{x}|\boldsymbol{z}_{\leq k})$ is powerful enough to reconstruct the observation $\boldsymbol{x}$ well, the variational posteriors of higher-level latent variables $\boldsymbol{z}_{>k}$ will be optimized to be close to their priors, i.e., $q_\phi(\boldsymbol{z}_{>k}|\boldsymbol{x}) \approx p_\theta(\boldsymbol{z}_{>k})$, leading the representations learned by VAE at higher layers to be meaningless and cannot provide faithful summaries for $\boldsymbol{x}$, which is well-known as the phenomenon of "*posterior collapse*" or "*latent variable collapse*" [18].

To find the potential solutions to alleviating "*posterior collapse*", in the following, we reinterpret this phenomenon from the perspective of information theory by extending the findings in [18] to a

hierarchical VAE scenario. For ease of undertanding, we define the mutual information between the data $\boldsymbol{x}$ and the higher-level latent variables $\boldsymbol{z}_{>k}$ as

$$\mathcal{I}_q(\boldsymbol{x}, \boldsymbol{z}_{>k}) = -\mathcal{H}_q(\boldsymbol{z}_{>k}|\boldsymbol{x}) + \mathcal{H}_q(\boldsymbol{z}_{>k}) = \mathbb{E}_{p(\boldsymbol{x})q_\phi(\boldsymbol{z}_{>k}|\boldsymbol{x})} \log q_\phi(\boldsymbol{z}_{>k}|\boldsymbol{x}) - \mathbb{E}_{q_\phi(\boldsymbol{z}_{>k})} \log q_\phi(\boldsymbol{z}_{>k}),$$

which is induced by the variational posterior $q_\phi(\boldsymbol{z}_{>k}|\boldsymbol{x})$. Then KL term in Eq. (26) can be rewritten as

$$\begin{aligned}
&\mathbb{E}_{p(\boldsymbol{x})}\left[\sum\nolimits_{l=1}^{L} D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_l|\boldsymbol{z}_{l+1})||p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{l+1}))\right] \\
=&\mathbb{E}_{p(\boldsymbol{x})}\left[\sum\nolimits_{l=1}^{k} D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_l|\boldsymbol{z}_{l+1})||p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{l+1}))\right] + \mathbb{E}_{p(\boldsymbol{x})}\left[D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_{>k}|\boldsymbol{x})||p_\theta(\boldsymbol{z}_{>k}))\right] \\
=&\mathbb{E}_{p(\boldsymbol{x})}\left[\sum\nolimits_{l=1}^{k} D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_l|\boldsymbol{z}_{l+1})||p_\theta(\boldsymbol{z}_l|\boldsymbol{z}_{l+1}))\right] + \mathcal{I}_q(\boldsymbol{x}, \boldsymbol{z}_{>k}) + D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_{>k})||p_\theta(\boldsymbol{z}_{>k})),
\end{aligned} \tag{27}$$

where $q_\phi(\boldsymbol{z}_{>k}) = \mathbb{E}_{p(\boldsymbol{x})}\left[q_\phi(\boldsymbol{z}_{>k}|\boldsymbol{x})\right]$. By substituting Eq. (27) into Eq. (26), due to the non-negativity of mutual information and KL divergence, we can find that maximizing the ELBO is opposite to maximizing the mutual information $\mathcal{I}_q(\boldsymbol{x}, \boldsymbol{z}_{>k})$. When $\mathcal{I}_q(\boldsymbol{x}, \boldsymbol{z}_{>k})$ is minimized to zero, the variational posterior $q_\phi(\boldsymbol{z}_{>k}|\boldsymbol{x})$ will be independent of the data $\boldsymbol{x}$, which leads to the phenomenon of "*posterior collapse*".

Thus, exploiting the property of the deep NTMs and building the skip connection between the $\boldsymbol{x}_n$ and $\boldsymbol{\theta}_n^{(l)}$ with two projection variants, as shown in Fig. 8, is promising to force the higher layers of latent variables be more informative by maximizing the mutual information.

K Evaluations of topic quality under more metrics

For evaluating the quality of the learned topics, we introduce more metrics for comparison between SawETM and dc-ETM-β (policy) (for brevity, we note it as "dc-ETM") including: topic diversity, topic coherence, and topic quality (a product of the topic diversity and topic coherence). As shown in Table 6, Table 7, and Table 8, dc-ETM can achieve comparable performance on the aspect of topic quality at the first layer while significantly outperform SawETM in the higher layers.

R8						
Metric	Diversity		Coherence		Topic Quality	
Layer #i	SawETM	dc-ETM	SawETM	dc-ETM	SawETM	dc-ETM
1	42.11	46.56	43.29	48.44	0.182	0.225
2	15.50	37.38	44.68	41.29	0.069	0.154
3	13.75	36.48	45.05	40.18	0.062	0.146
4	12.34	34.09	47.78	47.25	0.059	0.161
5	20.00	35.43	40.89	50.79	0.082	0.179
Average	20.70	37.98	44.34	45.59	0.091	0.173

Table 6: Topic Diversity, Coherence, and Quality (product of diversity and coherence) for each layer of SawETM and dc-ETM on R8 dataset.

20News						
Metric	Diversity		Coherence		Topic Quality	
Layer #i	SawETM	dc-ETM	SawETM	dc-ETM	SawETM	dc-ETM
1	38.82	33.05	31.01	32.32	0.120	0.107
2	22.53	40.04	55.91	54.69	0.126	0.218
3	9.531	53.75	63.67	56.21	0.067	0.302
4	9.310	47.96	59.42	60.77	0.055	0.291
5	7.309	53.02	62.04	63.23	0.045	0.335
Average	17.50	45.56	54.51	53.44	0.083	0.251

Table 7: Topic Diversity, Coherence, and Quality (product of diversity and coherence) for each layer of SawETM and dc-ETM on 20News dataset.

RCV1						
Metric	Diversity		Coherence		Topic Quality	
Layer #i	SawETM	dc-ETM	SawETM	dc-ETM	SawETM	dc-ETM
1	59.74	66.01	38.30	36.04	0.229	0.237
2	49.45	53.01	39.58	39.66	0.196	0.210
3	20.01	57.03	34.08	44.17	0.068	0.251
4	14.84	55.15	31.27	45.79	0.046	0.253
5	21.87	60.93	30.94	48.55	0.067	0.295
Average	33.18	58.42	34.83	42.84	0.121	0.249

Table 8: Topic Diversity, Coherence, and Quality (product of diversity and coherence) for each layer of SawETM and dc-ETM on R8 dataset.

L Layer-wise comparison under the clustering task

We provide an additional evaluation for the clustering tasks on each layer of SawETM and dc-ETM-β(policy) (for brevity, we note it as "dc-ETM"). As shown in Table 9, the higher layer $\boldsymbol{\theta}_n^{(l)}$ of SawETM can only preserve limited data information for the downstream document clustering task, which could explain why the SawETM cannot obtain a gain of performance improvement even if concatenating all hidden layers, specifically $\{\boldsymbol{\theta}_n^{(l)}\}_{l=1}^{L}$ (denoted as "All").

Datasets	20News				R8			
Metric	Purity		NMI		Purity		NMI	
Layer #i	SawETM	dc-ETM	SawETM	dc-ETM	SawETM	dc-ETM	SawETM	dc-ETM
1	43.33	40.11	50.77	44.12	75.25	71.30	42.97	38.34
2	33.52	42.22	45.88	46.72	76.01	75.38	38.41	40.14
3	26.72	44.50	40.12	47.03	34.38	74.22	27.09	44.75
4	12.51	45.76	30.20	50.26	22.10	75.00	17.03	43.10
5	11.82	46.89	26.78	53.81	20.17	76.19	15.20	45.89
All	38.69	48.60	39.33	55.79	75.89	78.29	39.55	48.62

Table 9: Comparison of different layers' $\boldsymbol{\theta}_n^{(l)}$ quality under the clustering task for SawETM and dc-ETM.

M Visualization of topic embeddings learned by different variants

To investigate the effects of dc-ETM variants α, β, and $policy$, we provide the t-sne visualization of 5th-layer topic embeddings learned by these variants. Since the semantics of learned topics vary with the model, to avoid cheery-picking and keep fair, we visualize the whole embeddings of 16 topics at the 5th layer. Note that, the hyper-parameters of the variants for t-SNE visualization are all the same.

Comparing Fig. 9 (SawETM) with Fig. 10 (dc-ETM-α), it could be obviously seen that the 5th-layer topics learned by SawETM are similar and meaningless, which means that the introduced skip-connection in dc-ETM-α is helpful to learn meaningful topics at higher layers.

Comparing Fig. 11 (dc-ETM-α-policy) with Fig. 10 (dc-ETM-α), it could be seen that the learned topics are more distinguishable in dc-ETM-α-policy, which indicates the effect of the policy gradient training schema.

Comparing Fig. 12 (dc-ETM-β) with Fig. 10 (dc-ETM-α), it could be seen that the learned topics are more distinguishable in dc-ETM-β, which indicates the effect of the efficient projection method β.

In all, combining the projection method β and the policy gradient training schema, we could acquire a significantly meaningful higher layers' topic embedding in Fig. 13 than SawETM in Fig. 9.

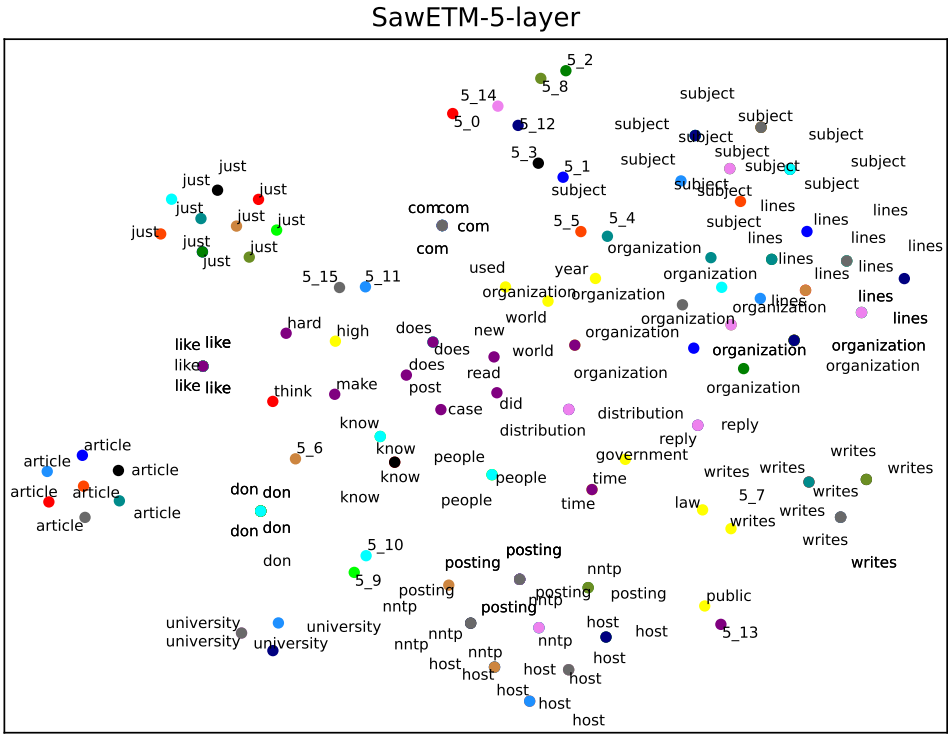

Figure 9: t-SNE visualization of the 5th-layer topic embeddings learned by SawETM.

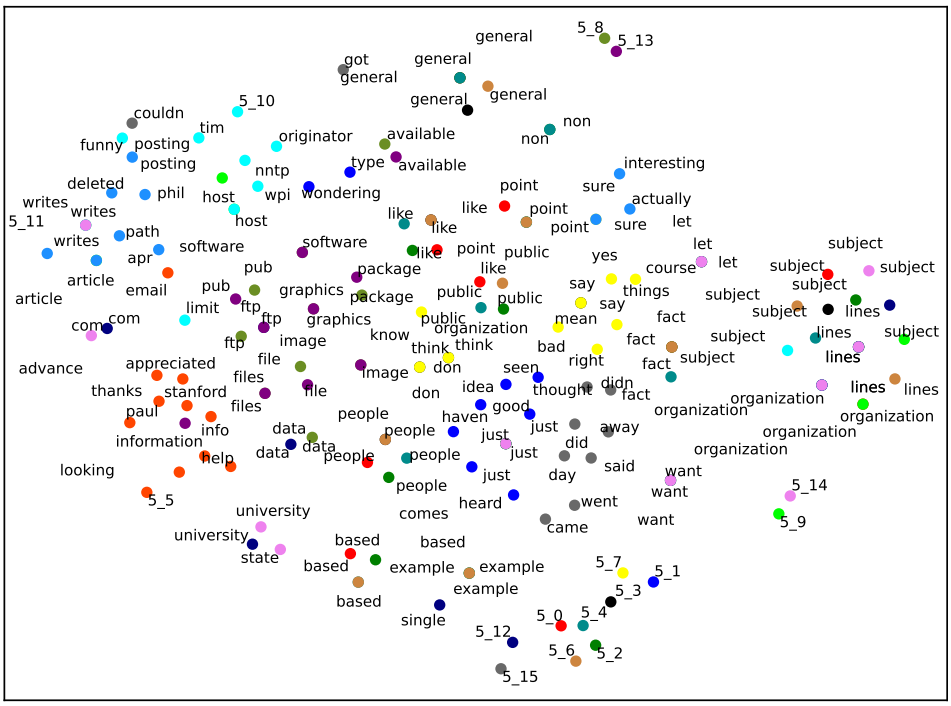

Figure 10: t-SNE visualization of the 5th-layer topic embeddings learned by dc-ETM-α.

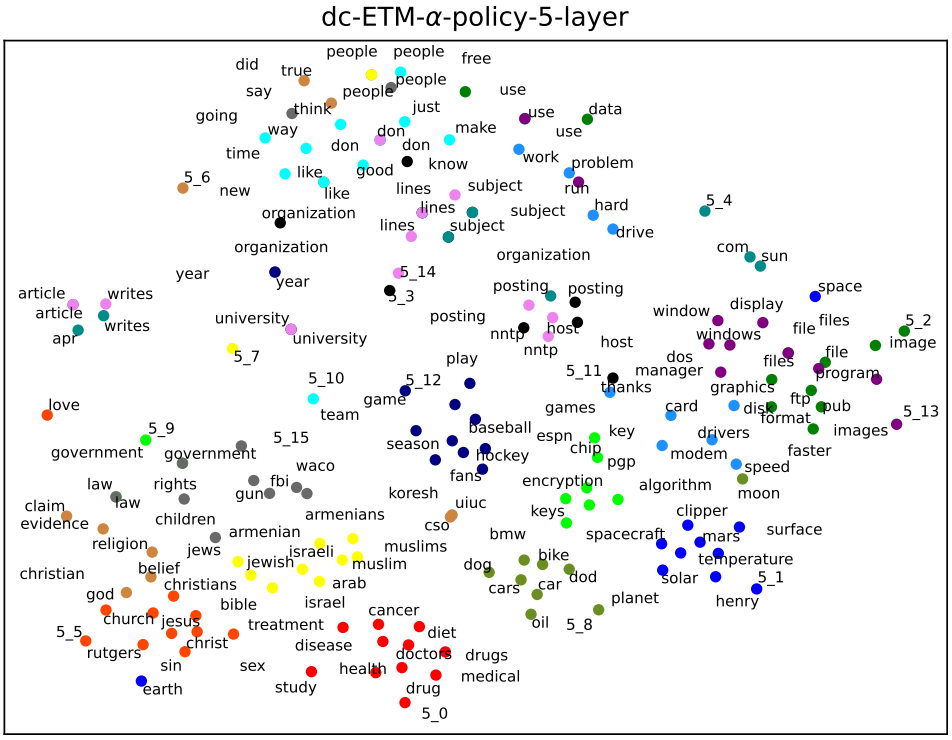

Figure 11: t-SNE visualization of the 5th-layer topic embeddings learned byf dc-ETM-α-policy.

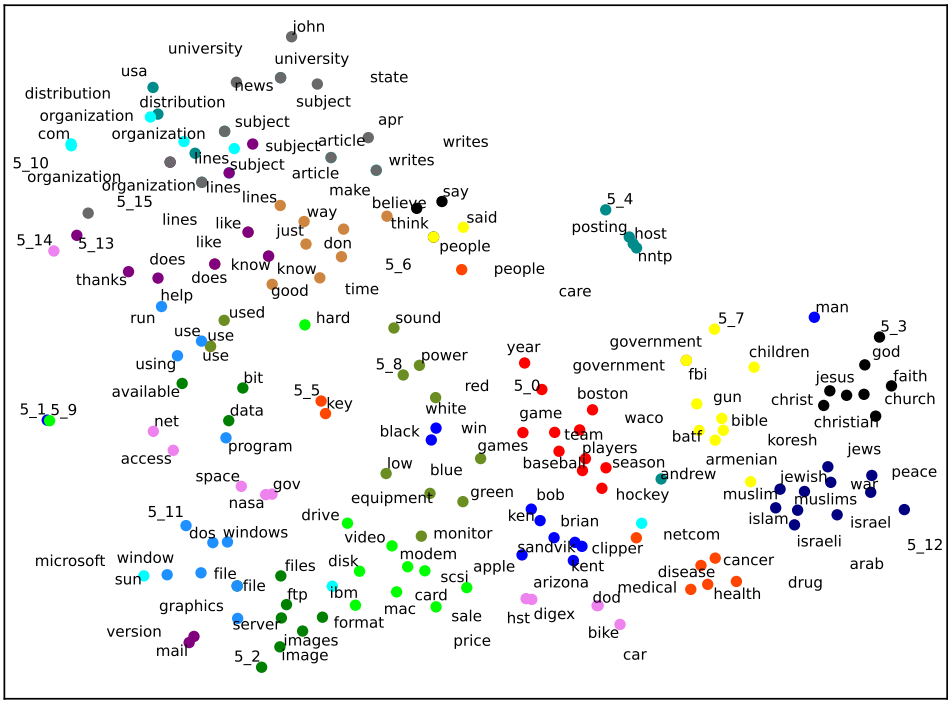

Figure 12: t-SNE visualization of the 5th-layer topic embeddings learned by dc-ETM-β.

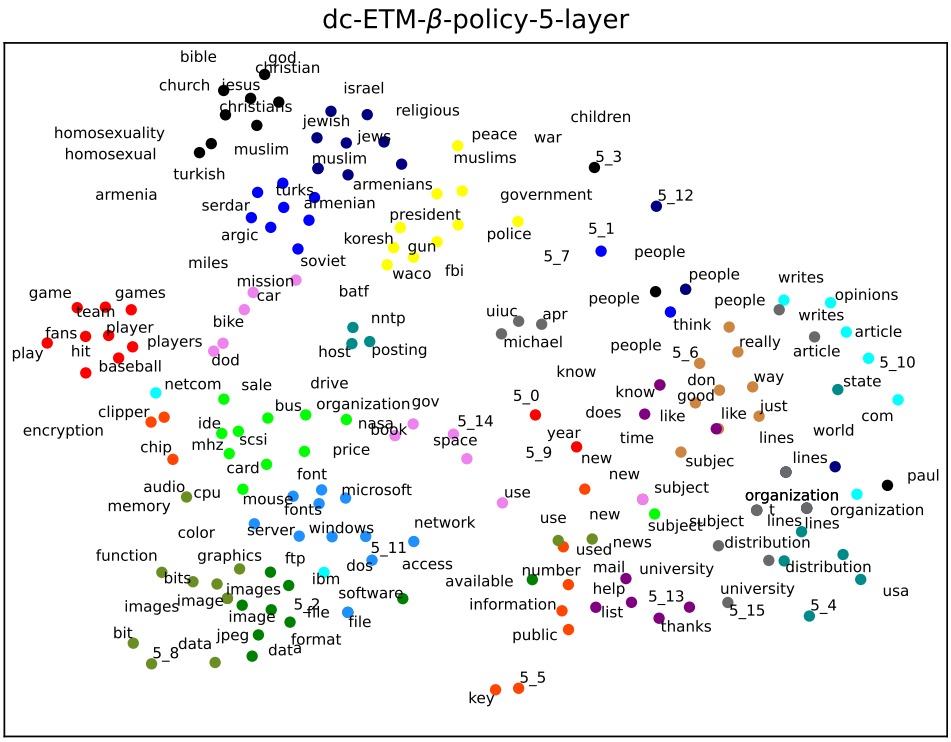

Figure 13: t-SNE visualization of the 5th-layer topic embeddings learned by dc-ETM-β-policy.