A Appendix: GO gradient for $\left\{\nabla_{\boldsymbol{\eta}^{(l)}}\mathcal{L}\right\}_{l=1}^{L}$

Except for the parameters $\{\boldsymbol{\eta}^{(l)}\}_{l=1}^{L}$ of variational distribution $q(\boldsymbol{\Phi}^{(l)}) \sim \mathrm{Dir}(\exp(\boldsymbol{\eta}^{(l)})), l = 1, \cdots, L$, the gradients of other parameters w.r.t. $\mathcal{L}$ in (11) can be easily calculated by standard BP algorithm. For the gradient of $\{\boldsymbol{\eta}^{(l)}\}_{l=1}^{L}$ w.r.t. $\mathcal{L}$, standard BP chain can not be applied straightforward, since Dirichlet distribution can not be reparameterized easily. In order to calculate gradient of $\{\boldsymbol{\eta}^{(l)}\}_{l=1}^{L}$ w.r.t. $\mathcal{L}$ with low variance, we apply GO gradient algorithm (Cong et al., 2019) here, which will be discussed in this Appendix.

With the following meanfield assumption:

$$q\left(\left\{\boldsymbol{\Phi}^{(l)}\right\}_{l=1}^{L}\right) = \prod_{l=1}^{L}\prod_{k=1}^{K_l} q_{\boldsymbol{\eta}_{:k}^{(l)}}\left(\boldsymbol{\Phi}_{:k}^{(l)}\right), \tag{12}$$

where K_l denotes the total numbers of topics at layer l of PGBN, $\boldsymbol{\Phi}_{:k}^{(l)}$ denotes the k-th column of $\boldsymbol{\Phi}^{(l)}$, and $\boldsymbol{\eta}_{:k}^{(l)}$ denotes the k-th column of $\boldsymbol{\eta}^{(l)}$. Therefore, we only need to discuss how to use GO to calculate the gradient $\nabla_{\boldsymbol{\eta}_{:k}^{(l)}}\mathbb{E}_{q_{\boldsymbol{\eta}_{:k}^{(l)}}(\boldsymbol{\Phi}_{:k}^{(l)})}\left[g(\boldsymbol{\Phi}_{:k}^{(l)})\right]$, where g is a function that gets rid of the expectation in (11). For simplicity, we use $\boldsymbol{\eta} \sim \mathbb{R}^{O\times 1}$ to denote $\boldsymbol{\eta}_{:k}^{(l)}$, and $\boldsymbol{\phi} \sim \mathbb{R}^{O\times 1}$ to denote $\boldsymbol{\Phi}_{:k}^{(l)}$. With these illustrations, our core problem is defined as how to calculate

$$\nabla_{\boldsymbol{\eta}}\mathbb{E}_{q_{\boldsymbol{\eta}}(\boldsymbol{\phi})}\left[g(\boldsymbol{\phi})\right] \tag{13}$$

As we know, if $q(\boldsymbol{\phi}) \sim \mathrm{Dir}(\exp(\boldsymbol{\eta}))$, we can sample it as

$$\begin{aligned}&\boldsymbol{\psi}_o \sim \mathrm{Gam}(\exp(\boldsymbol{\eta}_o), 1), o = 1, \cdots, O\\ &\boldsymbol{\phi} = \left[\frac{\boldsymbol{\psi}_1}{\sum_{o=1}^{O}\boldsymbol{\psi}_o}; \cdots; \frac{\boldsymbol{\psi}_O}{\sum_{o=1}^{O}\boldsymbol{\psi}_o}\right].\end{aligned} \tag{14}$$

As a result, the core problem in (13) is changed to

$$\nabla_{\boldsymbol{\eta}}\mathbb{E}_{q_{\boldsymbol{\eta}}(\boldsymbol{\psi})}\left[g(\boldsymbol{\psi})\right]. \tag{15}$$

As given in Theorem 1 in Cong et al. (2019), the above gradient can we written as

$$\nabla_{\boldsymbol{\eta}}\mathbb{E}_{q_{\boldsymbol{\eta}}(\boldsymbol{\psi})}\left[g(\boldsymbol{\psi})\right] = \mathbb{E}_{q_{\boldsymbol{\eta}}(\boldsymbol{\psi})}\left[\mathbb{G}_{\boldsymbol{\eta}}^{q_{\boldsymbol{\eta}}(\boldsymbol{\psi})}\mathbb{D}_{\boldsymbol{\psi}}\left[g(\boldsymbol{\psi})\right]\right], \tag{16}$$

where

$$\mathbb{D}_{\boldsymbol{\psi}}\left[g(\boldsymbol{\psi})\right] = \left[\mathbb{D}_{\boldsymbol{\psi}_1}[g(\boldsymbol{\psi})], \cdots, \mathbb{D}_{\boldsymbol{\psi}_o}[g(\boldsymbol{\psi})], \cdots, \mathbb{D}_{\boldsymbol{\psi}_O}[g(\boldsymbol{\psi})]\right]^T$$

with

$$\mathbb{D}_{\boldsymbol{\psi}_o}[g(\boldsymbol{\psi})] \overset{def}{=} \nabla_{\boldsymbol{\psi}_o}g(\boldsymbol{\psi}) \tag{17}$$

which is easy to calculate by stand BP algorithm;

$$\mathbb{G}_{\boldsymbol{\eta}}^{q_{\boldsymbol{\eta}}(\boldsymbol{\psi})} \overset{def}{=} \left[s_{\boldsymbol{\eta}}^{q(\boldsymbol{\psi}_1)}, \cdots, s_{\boldsymbol{\eta}}^{q(\boldsymbol{\psi}_o)}, \cdots, s_{\boldsymbol{\eta}}^{q(\boldsymbol{\psi}_O)}\right], o = 1, \cdots, O,$$

with

$$\begin{aligned}s_{\boldsymbol{\eta}}^{q(\boldsymbol{\psi}_o)} &= \frac{-1}{q(\boldsymbol{\psi}_o)}\nabla_{\boldsymbol{\eta}}Q(\boldsymbol{\psi}_o)\\ &= \frac{-1}{q(\boldsymbol{\psi}_o)}\left[\nabla_{\boldsymbol{\eta}_1}Q(\boldsymbol{\psi}_o), \cdots, \nabla_{\boldsymbol{\eta}_O}Q(\boldsymbol{\psi}_o)\right] \in \mathbb{R}^{O\times 1},\end{aligned} \tag{18}$$

where $Q(\boldsymbol{\psi}_o)$ is the CDF of $q(\boldsymbol{\psi}_o)$. As discussed in (14), $q(\boldsymbol{\psi}_o) \sim \mathrm{Gam}(\exp(\boldsymbol{\eta}_o), 1)$, and each $\{\boldsymbol{\psi}_o\}_{o=1}^{O}$ are independently sampled. Thus, the elements in $s_{\boldsymbol{\eta}}^{q(\boldsymbol{\psi}_o)}$ are all zeros except the o-th element being $s_{\boldsymbol{\eta}_o}^{q(\boldsymbol{\psi}_o)} = \frac{-1}{q(\boldsymbol{\psi}_o)}\nabla_{\boldsymbol{\eta}_o}Q(\boldsymbol{\psi}_o)$. Thus, matrix $\mathbb{G}_{\boldsymbol{\eta}}^{q_{\boldsymbol{\eta}}(\boldsymbol{\psi})}$ is a diagonal matrix.

In Cong et al. (2019), if $q(\boldsymbol{\psi}_o)$ is the gamma distribution $\mathrm{Gam}(\alpha, 1)$ where $\alpha = \exp(\boldsymbol{\eta}_o)$, the authors give the result of $s_{\boldsymbol{\eta}_o}^{q(\boldsymbol{\psi}_o)}$ as

$$s_{\boldsymbol{\eta}_o}^{q(\boldsymbol{\psi}_o)} = \frac{[\log(\boldsymbol{\psi}_0) - \digamma(\alpha)]\Gamma(\alpha, \boldsymbol{\psi}_0) + \boldsymbol{\psi}_0\mathcal{T}(3, \alpha, \boldsymbol{\psi}_0)}{\boldsymbol{\psi}_0^{\alpha-1}\exp^{-\boldsymbol{\psi}_0}}, \tag{19}$$

where $\digamma(\cdot)$ is the digamma function, $\Gamma(\cdot,\cdot)$ is the upper incomplete gamma function, and $\mathcal{T}(\cdot,\cdot,\cdot)$ is a special case of Meijer G-function (Geddes et al., 1990).

For clearer to understand know how to calculate the gradient of $\left\{\nabla_{\boldsymbol{\eta}^{(l)}}\mathcal{L}\right\}_{l=1}^{L}$, as shown in Fig. 6, we give the illustration of the feed-forward and back-propagation process of parameters $\boldsymbol{\eta}$.

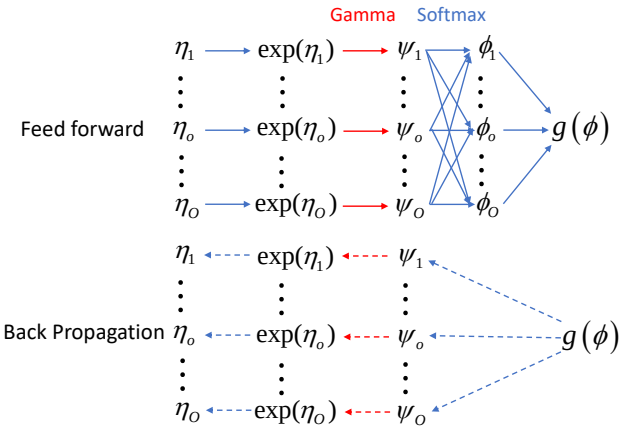

Figure 6: Feed forward and back propagation process of parameters $\boldsymbol{\eta}$, where for feed forward, the blue lines denote deterministic mapping and the red ones denote stochastic sample; for back propagation, the blue lines denote standard BP chain calculated by (17) and the red ones denote the BP calculated by (19).

Table 2: The statstics of each dataset after preprocessing.

Dataset	# Docs	# Training	# Test	# Word	# Nodes	# Classes	Average Length
20NG	18,846	11,314	7,532	42,757	62,051	20	221.26
R8	7,674	5,485	2,189	7,688	15,810	8	65.72
R52	9,100	6,532	2,568	8,892	18,440	52	69.82
Ohsumed	7,400	3,357	4,043	14,157	22,005	23	135.82
MR	10,662	7,108	3,554	18,764	29,874	2	20.39

B Appendix: The whole algorithm of our model

In this paper, different from many existing models (Yao et al., 2019; Liu et al., 2019; Chen et al., 2019; Li et al., 2019) that separate the graph construction and GCN learning, we propose a joint learning method to build graph dynamically with GCN learning based a unified loss function in (11). In Algorithm 1, we give a detailed step to illustrate how to update the models.

Algorithm 1 Joint learning of HTG and GCN.

Initialize WUDVE encoder parameters $\mathbf{W}_e$, GCN parameters $\{\mathbf{W}_G^{(l)}\}_{l=1}^2$, softmax function parameters $\mathbf{W}_c$ and $\mathbf{W}_c^{'}$, variational distribution parameters of topics $\{\boldsymbol{\eta}^{(l)}\}_{l=1}^L$.

for $iter = 1, 2, \cdots$ **do**

1) Randomly select a mini-batch containing $\tilde{N}$ documents with its labels $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^{\tilde{N}}$;

2) Draw random noise $\left\{\boldsymbol{\varepsilon}_n^{(l)}\right\}_{n=1,l=1}^{N,L}$ from uniform distribution to reparameterize the Weibull variational distribution of $\boldsymbol{\theta}_n$;

3) Approximate the expectation in $\mathcal{L}$ (11) by one sample;

4) Calculate $\nabla_{\mathbf{W}_e}\mathcal{L}$, $\nabla_{\mathbf{W}_G}\mathcal{L}$, $\nabla_{\mathbf{W}_c}\mathcal{L}$, $\nabla_{\mathbf{W}_c'}\mathcal{L}$ by standard Back Propagation (BP);

5) Calculate $\left\{\nabla_{\boldsymbol{\eta}^{(l)}}\mathcal{L}\right\}_{l=1}^{\mathcal{L}}$ according to GO gradient (Cong et al., 2019), which is specified in Appendix A.

6) Update $\mathbf{W}_e$,$\{\mathbf{W}_G^{(l)}\}_{l=1}^2$, $\mathbf{W}_c$, $\mathbf{W}_c^{'}$ and $\{\boldsymbol{\eta}^{(l)}\}_{l=1}^L$ through gradient descend algorithm such as ADAM to minimize the loss $-\mathcal{L}$ in (11)).

end for

C Detailed statistics of each dataset

For better understand the statistics of each dataset, the statistics of each dataset after preprocessing are summarized in Table 2. The number of nodes $|V| = D + \sum_{l=1}^3 K_l + N$, where D represents the number of words, K_l represents the number of topics at layer l (we use $K_1 = 256$, $K_2 = 128$, $K_3 = 64$ in all experiments), and N represents the number of training documents, respectively. Note that, in practice at each iteration in Algorithm 1, we only select a mini-batch containing $\tilde{N}$ documents with its labels. This operation can not break our developed HTG due to the fact that one document node only has edges with other type nodes but has no edge with other document nodes.

D Semantics among document-topic nodes of 20NG

We send each word embedding at GCN-layer-2 to the classifier $\mathbf{W}_c$ in (8). In Fig. 7, we show the top 10 words with highest values corresponding to some classes on 20NG. Clearly, we note that the top 10 words are interpretable, which are very close to the label's meaning.

E DHTG with different layers of topics

We use a well-trained GBN model to build a static HTG with different layers, represented as SHTG-L1, SHTG-L2, SHTG-L3, respectively. The comparison results are listed in Table 1. As consistently observed across all datasets, the classification accuracy increases with more layers, illustrating the effectiveness of multi-layer document representations. As a complementary experiment, we build a DHTG with different topic-layers, with the test accuracy results on five datasets listed in Table 3. A similar phenomenon is observed that the classification accuracy increases with more topic layers.

F Another Statistics of test accuracy compare between DHTG and textGCN

In Yao et al. (2019), the mean and standard deviation of the test accuracy is achieved by running 10-times experiments with different weight initialization, but the same training/test split. To make fair comparison with the results listed in Yao et al. (2019), we follow their setting and summarize the results as Table 1. A reviewer suggested us to show the mean and standard

rec.sport.baseball	sci.med	rec.autos	comp.windows.x	soc.religion.christian	talk.politics.guns
hitter	candida	car	windows	church	gun
pitching	disease	cars	dos	jesus	firearms
baseball	patients	v12	exe	christians	ax
braves	vitamin	callison	file	faith	fbi
pitcher	syndrome	engine	win3	bible	handheld
batting	infection	toyota	drivers	christianity	weapons
cubs	chronic	nissan	fonts	catholic	handgun
players	doctor	mustang	files	christian	firearm
phillies	clinical	wagon	font	heaven	amendment
pitchers	hiv	ford	zip	romans	handguns

Figure 7: Words with highest values for several classes in 20NG .

Table 3: Test classification accuracy on five datasets with different layers of PGBN in DHTG.

Model	20NG	R8	R52	Ohsumed	MR
DHTG-L1	86.69 ± 0.08	97.21 ± 0.07	93.67 ± 0.16	68.15 ± 0.40	77.02 ± 0.16
DHTG-L2	86.93 ± 0.07	97.29 ± 0.05	93.81 ± 0.13	68.51 ± 0.36	77.11 ± 0.15
DHTG-L3	87.13 ± 0.07	97.33 ± 0.06	93.93 ± 0.10	68.80 ± 0.33	77.21 ± 0.11

Table 4: Test accuracy of textGCN and DHTG on five dataset, where the mean and the standard deviation are achieved by running 10 times experiments with different weight initialization and different training/test split.

Model	20NG	R8	R52	Ohsumed	MR
textGCN	85.27 ± 0.36	96.51 ± 0.35	94.07 ± 0.32	68.56 ± 0.52	75.61 ± 0.31
DHTG	86.85 ± 0.23	97.10 ± 0.19	94.33 ± 0.28	69.08 ± 0.41	77.10 ± 0.19

deviation of the test accuracy by running 10-times experiments with different weight initialization and different training/test split. In Table 4, we give the corresponding results. Clearly, compared with textGCN, DHTG has higher mean and lower standard deviation, demonstrating the superior performance of DHTG.